The iMagine AI platform

WP4 update and status

Álvaro López García - Advanced Computing and e-Science Group (IFCA-CSIC-UC)
(aloga@ifca.unican.es)

iMagine RP1 review

December 5th 2023

Instituto de Física de Cantabria
CSIC Consejo Superior de Investigaciones Científicas
UC

DEEP-HDC, AI4EOSC, iMagine, AI4OS…

- **DEEP-1**, **DEEP-2**: Platform releases
- Platform and software tightly coupled and interlinked, difficult to self-deploy and customize

- **AI4EOSC platform** → Platform “powered by AIOS”
  - DEEP-3 → AI4EOSC-3
- **AI4OS** → software distribution
  - Possible to build custom platforms, partially integrated with AI4EOSC platform (i.e. reusing services) or not
  - https://github.com/AI4OS

- **iMagine AI platform**
  - Customized platform for AI image processing
  - Support for AI service deployment and creation
  - Exploitation if DEEP/AI4OS software as technology provider

DEEP-HDC, AI4EOSC, iMagine, AI4OS...

- **DEEP-1**, **DEEP-2**: Platform releases
- Platform and software tightly coupled and interlinked, difficult to self-deploy and customize

(large) User input, missing features

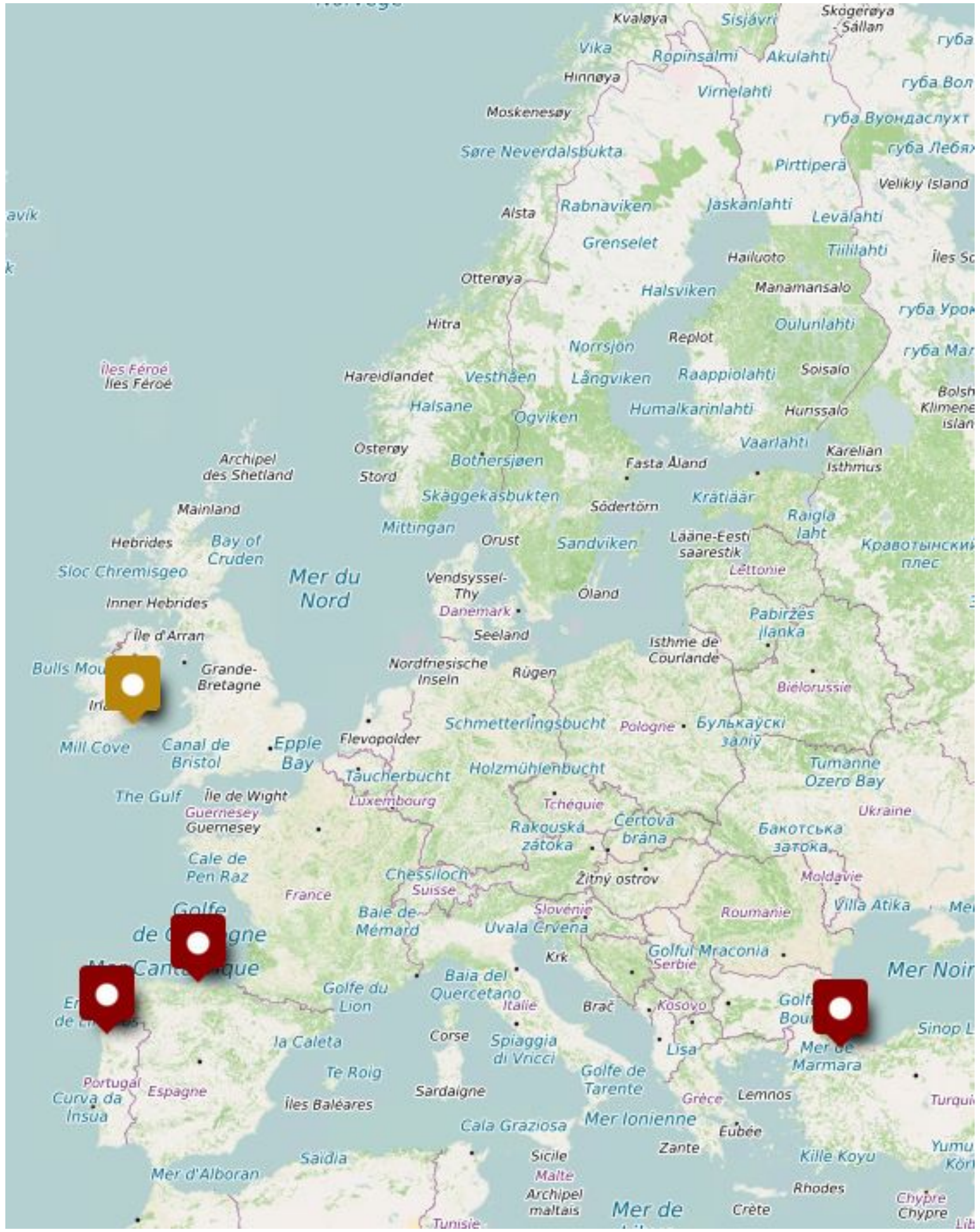
- **AI4EOSC platform** → Platform “powered by AIOS”
  - DEEP-3 → AI4EOSC-3
- **AI4OS** → software distribution
  - Possible to build custom platforms, partially integrated with AI4EOSC platform (i.e. reusing services) or not
  - https://github.com/AI4OS

Feature delivery

- **iMagine AI platform**
  - Customized platform for AI image processing
  - Support for AI service deployment and creation
  - Exploitation if DEEP/AI4OS software as technology provider

iMagine AI platform customizations

A distributed and federated platform



- Integration of different cloud computing providers from the EGI Federated Cloud
  - GPU access
    - CSIC Scientific Cloud (es), INCD Cloud (pt), TÜBITAK-ULAKBIM (tk)
  - CPU access
    - Walton Institute (ie)

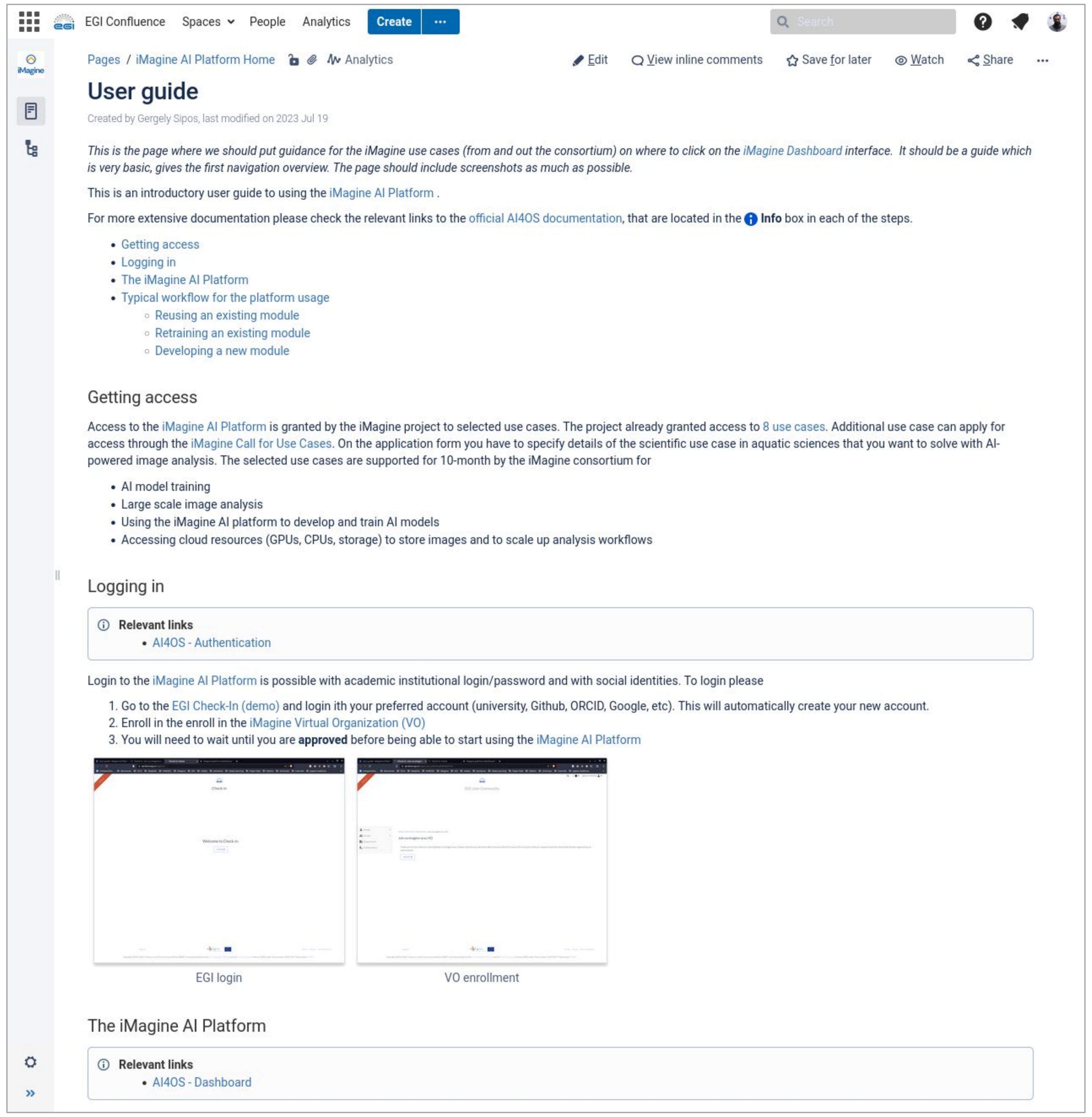
- Transparent access to pan-EU e-Infrastructures state of the art resources
  - Distributed access transparent for users
  - Integration with EOSC ecosystem

- Not imposing any special libraries or hard requirements for users
  - Only requirement is (light) integration with API

- Compatible with KServe community API for model delivery (in progress)

User engagement: how we deliver?

- EOSC portal onboarding
  - https://marketplace.eosc-portal.eu/services/imaging-ai-platform-for-aquatic-science
- Webinars, user meetings
  - iMagine Competence Centre workshop: https://indico.egi.eu/event/5999/
  - EGI Conference 2023: https://whova.com/web/M8zkrnLo5DUwlnug54VINPkHTdssyI49PHa2OjCW2Qg%3D/Agenda/
  - AI4EOSC + iMagine user workshop: https://indico.scc.kit.edu/event/3845/
- Comprehensive documentation
  - https://docs.ai4os.eu/
- Customized user guide and quickstart
  - https://confluence.egi.eu/display/IMPAIP/User+guide



User guide

Created by Gergely Sipos, last modified on 2023 Jul 19

This is the page where we should put guidance for the iMagine use cases (from and out the consortium) on where to click on the iMagine Dashboard interface. It should be a guide which is very basic, gives the first navigation overview. The page should include screenshots as much as possible.

This is an introductory user guide to using the iMagine AI Platform .

For more extensive documentation please check the relevant links to the official AI4OS documentation, that are located in the **Info** box in each of the steps.

- Getting access
- Logging in
- The iMagine AI Platform
- Typical workflow for the platform usage
  - Reusing an existing module
  - Retraining an existing module
  - Developing a new module

Getting access

Access to the iMagine AI Platform is granted by the iMagine project to selected use cases. The project already granted access to 8 use cases. Additional use case can apply for access through the iMagine Call for Use Cases. On the application form you have to specify details of the scientific use case in aquatic sciences that you want to solve with AI-powered image analysis. The selected use cases are supported for 10-month by the iMagine consortium for

- AI model training
- Large scale image analysis
- Using the iMagine AI platform to develop and train AI models
- Accessing cloud resources (GPUs, CPUs, storage) to store images and to scale up analysis workflows

Logging in

Relevant links
- AI4OS - Authentication

Login to the iMagine AI Platform is possible with academic institutional login/password and with social identities. To login please

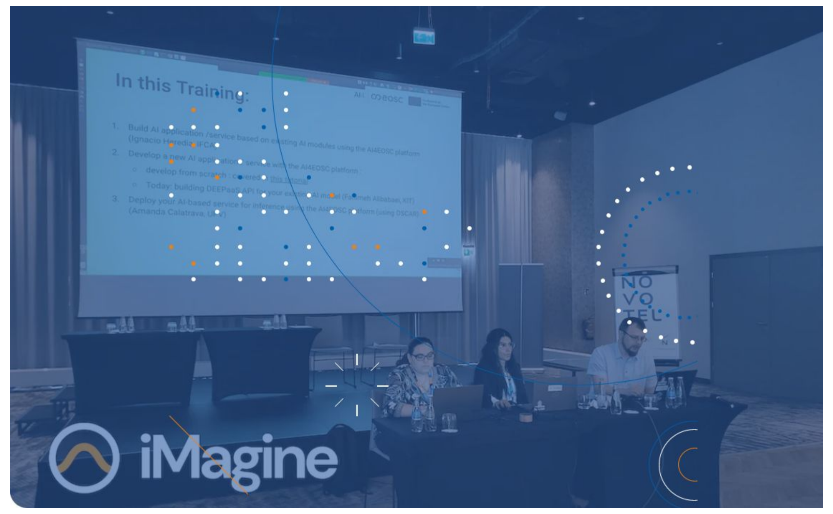
1. Go to the EGI Check-In (demo) and login ith your preferred account (university, Github, ORCID, Google, etc). This will automatically create your new account.
2. Enroll in the enroll in the iMagine Virtual Organization (VO)
3. You will need to wait until you are **approved** before being able to start using the iMagine AI Platform

EGI login

VO enrollment

The iMagine AI Platform

Relevant links
- AI4OS - Dashboard

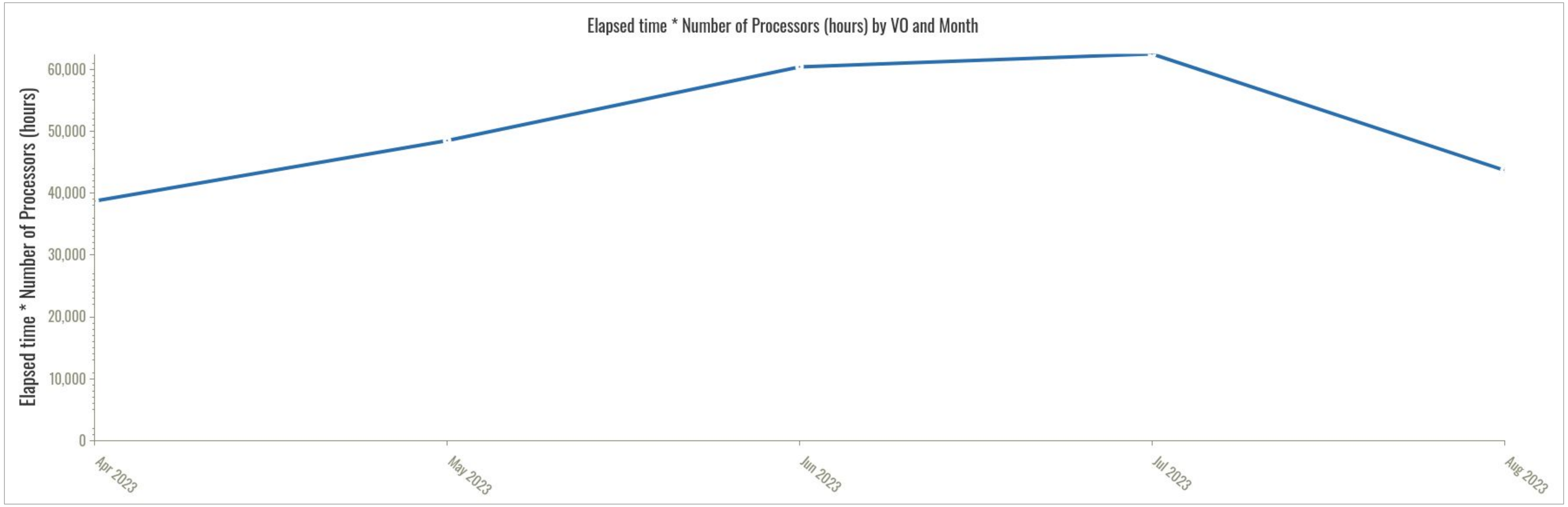


Platform status

Integration status

Federated Compute Infrastructure (T4.3)

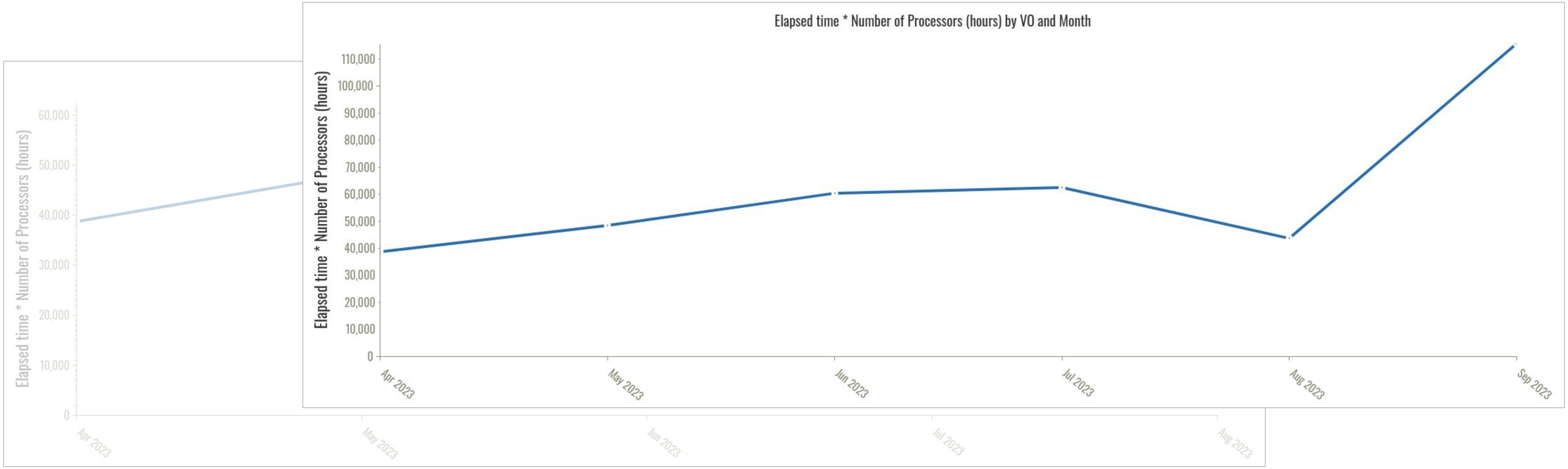
- All sites now supporting iMagine VO (Operators group) to deploy platform services
- Usage reported through EGI accounting system (link)



Integration status

Federated Compute Infrastructure (T4.3)

- All sites now supporting iMagine VO (Operators group) to deploy platform services
- Usage reported through EGI accounting system (link)



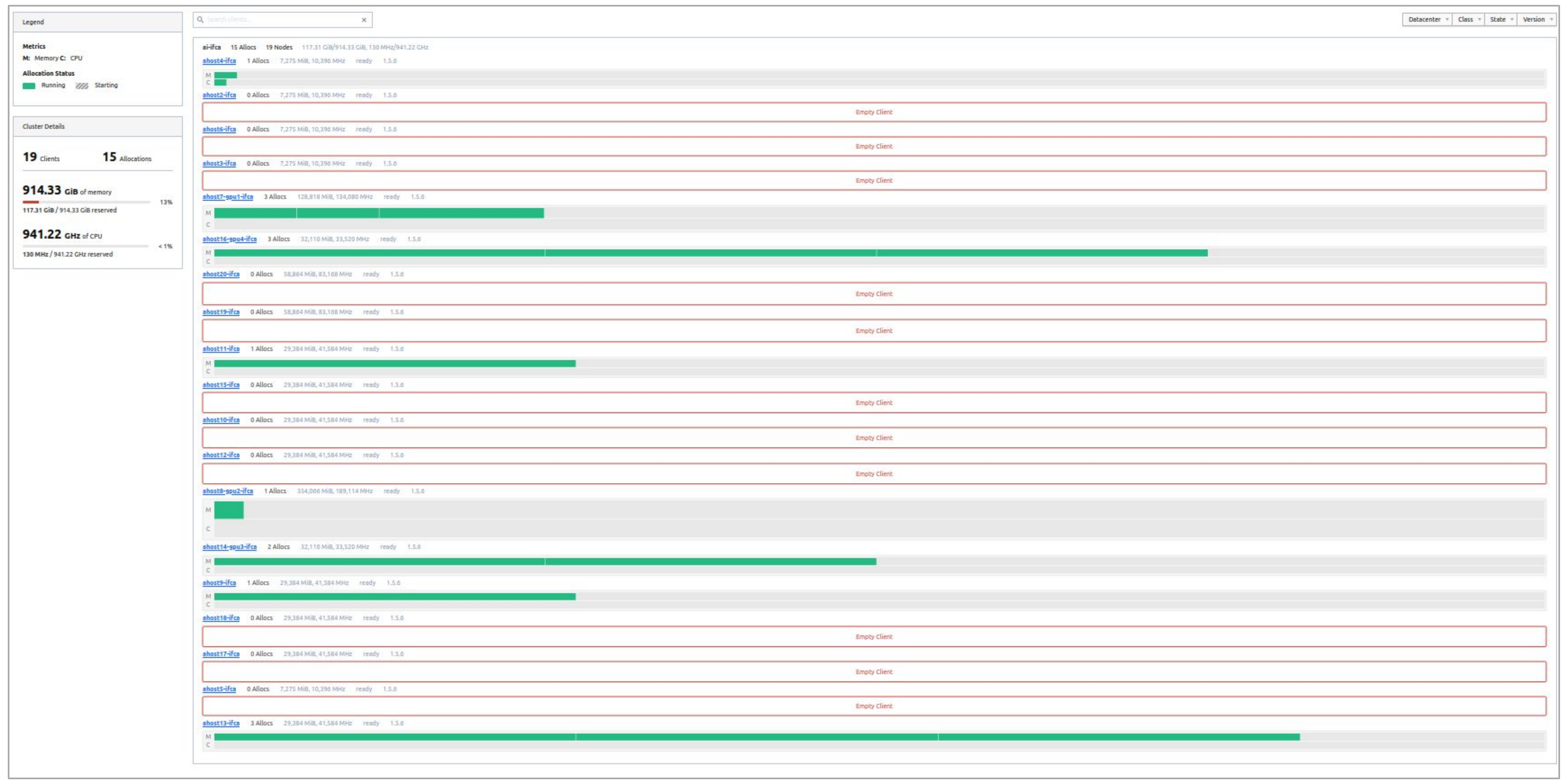
Integration status

*Imagine AI **development** platform (T4.1)*

- Production system transitioned from DEEP software stack to AI4OS
    - Now running fully in AI4OS
    - EGI CheckIn integration

- Production system
    - CSIC, INCD and TUBITAK (GPU)
    - Integrating Walton (CPU)

- Development testbed
    - CSIC, INCD



Legend

Metrics
M: Memory C: CPU

Allocation Status
🟩 Running ▨ Starting

Cluster Details

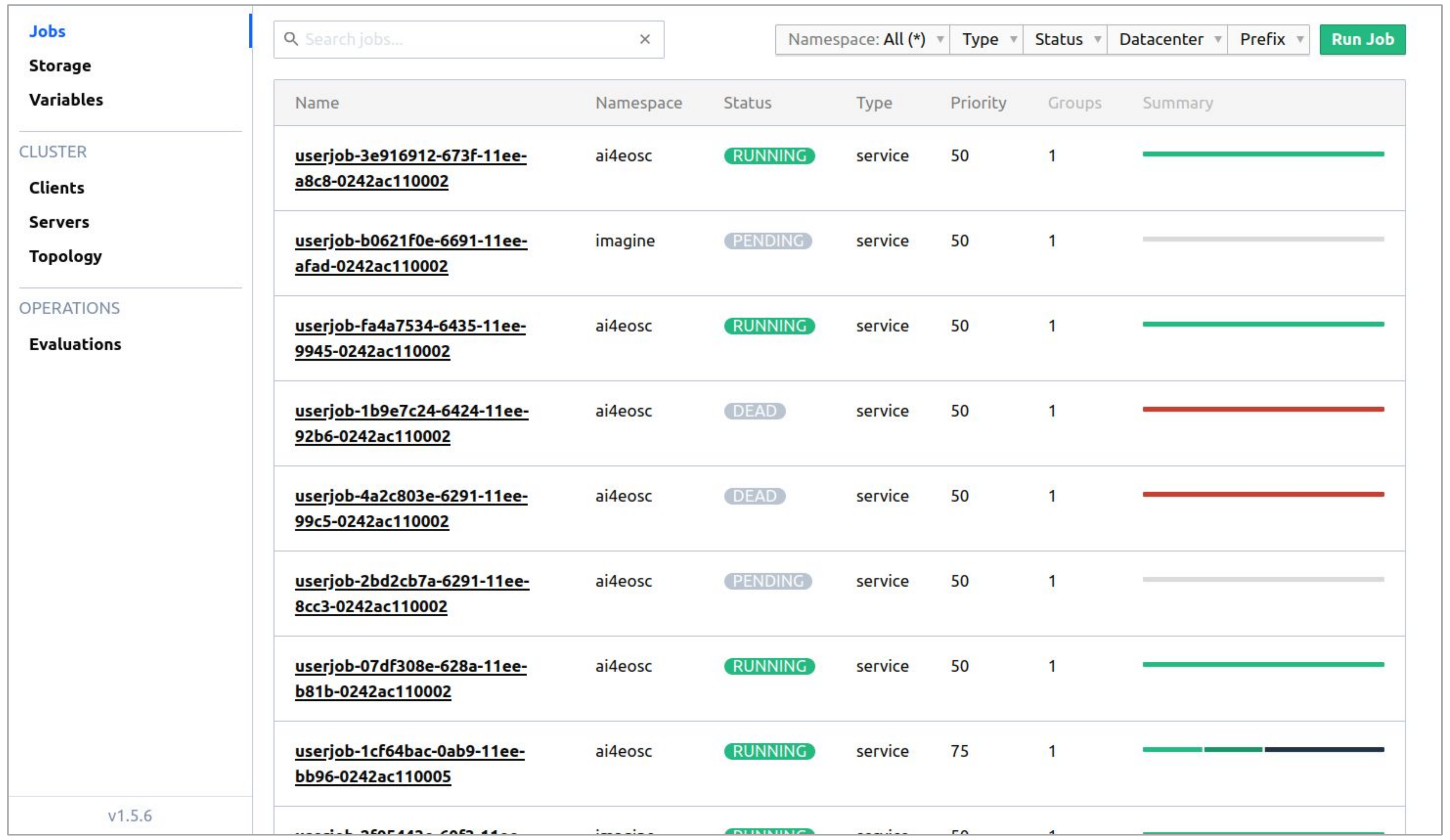
19 Clients 15 Allocations

914.33 GiB of memory 13%
117.31 GiB / 914.33 GiB reserved

941.22 GHz of CPU < 1%
130 MHz / 941.22 GHz reserved

ai-ifca 15 Allocs 19 Nodes 117.31 GiB/914.33 GiB, 130 MHz/941.22 GHz

ahost4-ifca	1 Allocs	7,275 MiB, 10,396 MHz	ready	1.5.6
ahost2-ifca	0 Allocs	7,275 MiB, 10,396 MHz	ready	1.5.6
ahost6-ifca	0 Allocs	7,275 MiB, 10,396 MHz	ready	1.5.6
ahost3-ifca	0 Allocs	7,275 MiB, 10,396 MHz	ready	1.5.6
ahost7-gpu1-ifca	3 Allocs	128,818 MiB, 134,080 MHz	ready	1.5.6
ahost16-gpu4-ifca	3 Allocs	32,110 MiB, 33,520 MHz	ready	1.5.6
ahost20-ifca	0 Allocs	58,864 MiB, 83,168 MHz	ready	1.5.6
ahost19-ifca	0 Allocs	58,864 MiB, 83,168 MHz	ready	1.5.6
ahost11-ifca	1 Allocs	29,384 MiB, 41,584 MHz	ready	1.5.6
ahost15-ifca	0 Allocs	29,384 MiB, 41,584 MHz	ready	1.5.6
ahost10-ifca	0 Allocs	29,384 MiB, 41,584 MHz	ready	1.5.6
ahost12-ifca	0 Allocs	29,384 MiB, 41,584 MHz	ready	1.5.6
ahost8-gpu2-ifca	1 Allocs	354,066 MiB, 189,114 MHz	ready	1.5.6
ahost14-gpu3-ifca	2 Allocs	32,110 MiB, 33,520 MHz	ready	1.5.6
ahost9-ifca	1 Allocs	29,384 MiB, 41,584 MHz	ready	1.5.6
ahost18-ifca	0 Allocs	29,384 MiB, 41,584 MHz	ready	1.5.6
ahost17-ifca	0 Allocs	29,384 MiB, 41,584 MHz	ready	1.5.6
ahost5-ifca	0 Allocs	7,275 MiB, 10,396 MHz	ready	1.5.6
ahost13-ifca	3 Allocs	29,384 MiB, 41,584 MHz	ready	1.5.6



Name	Namespace	Status	Type	Priority	Groups	Summary
userjob-3e916912-673f-11ee-a8c8-0242ac110002	ai4eosc	RUNNING	service	50	1	
userjob-b0621f0e-6691-11ee-afad-0242ac110002	imagine	PENDING	service	50	1	
userjob-fa4a7534-6435-11ee-9945-0242ac110002	ai4eosc	RUNNING	service	50	1	
userjob-1b9e7c24-6424-11ee-92b6-0242ac110002	ai4eosc	DEAD	service	50	1	
userjob-4a2c803e-6291-11ee-99c5-0242ac110002	ai4eosc	DEAD	service	50	1	
userjob-2bd2cb7a-6291-11ee-8cc3-0242ac110002	ai4eosc	PENDING	service	50	1	
userjob-07df308e-628a-11ee-b81b-0242ac110002	ai4eosc	RUNNING	service	50	1	
userjob-1cf64bac-0ab9-11ee-bb96-0242ac110005	ai4eosc	RUNNING	service	75	1	

Integration status

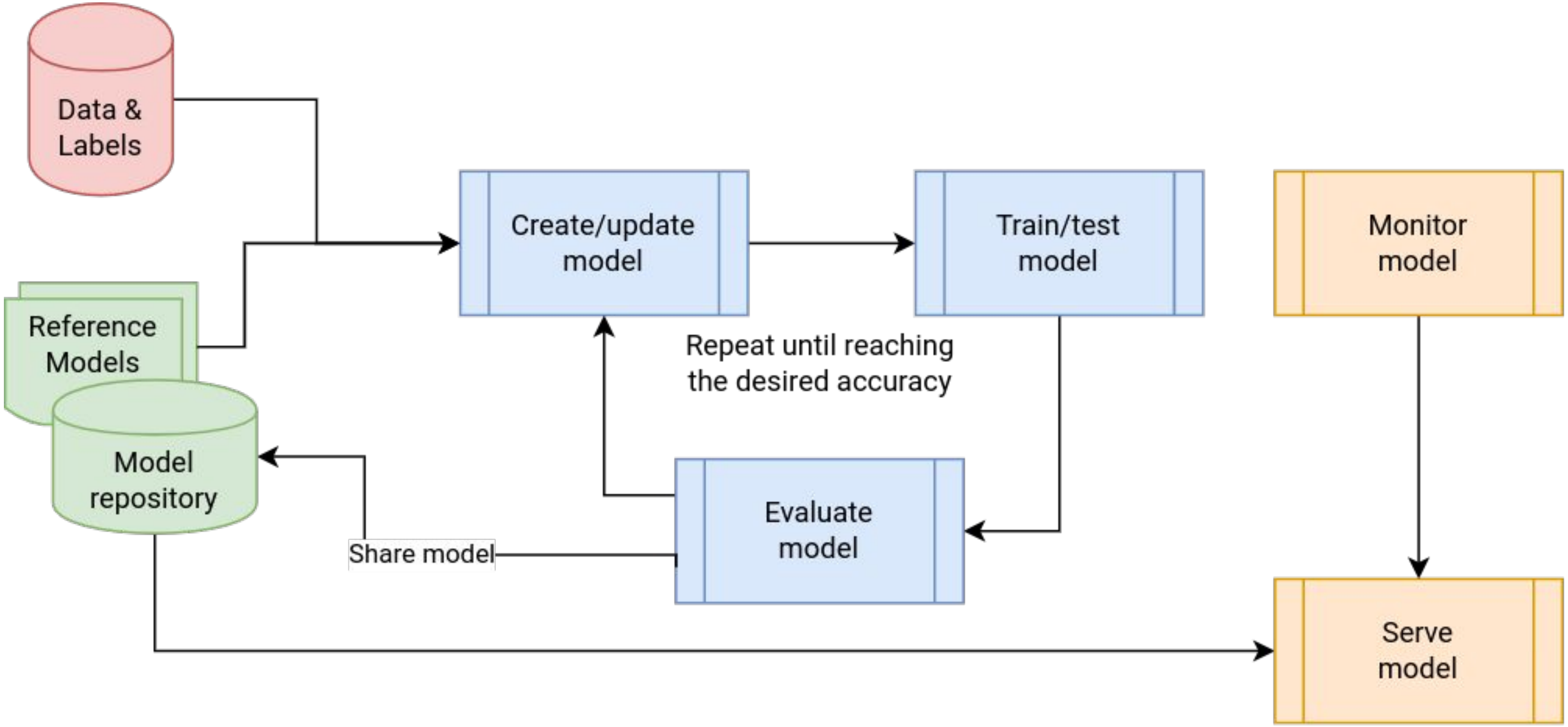
*Imagine AI **service** platform (T4.2)*

- Preliminary integration work, bulk work not yet started
  - Depending on WP5 work to deliver
- Exploiting CPU resources for inference
- Different approaches for deployment of services
  - Standalone (i.e. single server in a VM through EGI IM)
  - OSCAR clusters (Kubernetes based, serverless inference, through EGI IM)
  - Platform-level preview services (self-contained in platform)

For end users and user communities

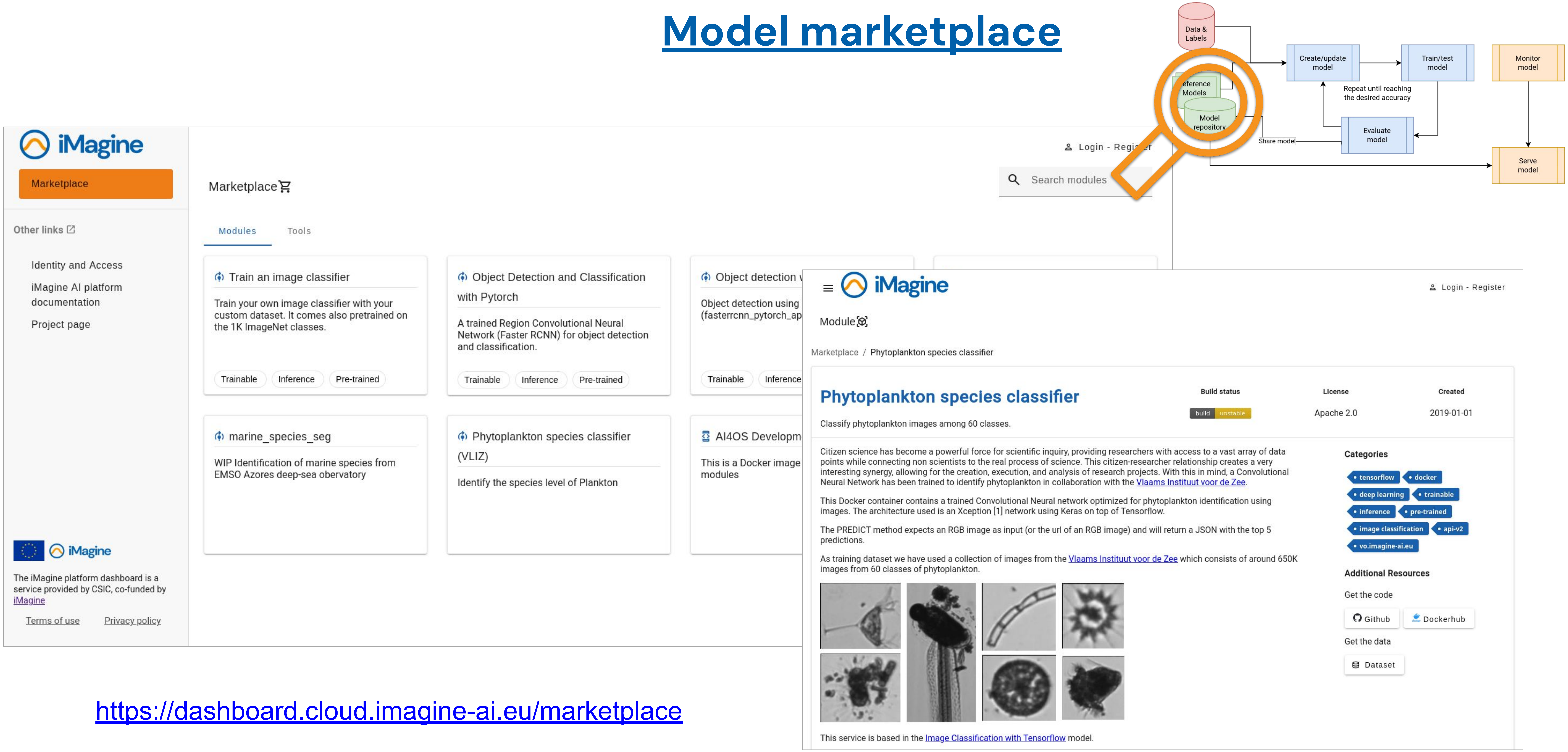
Platform features

AI/ML application development lifecycle



Services for AI/ML development

Model marketplace



iMagine

Marketplace

Other links

Identity and Access

iMagine AI platform documentation

Project page

The iMagine platform dashboard is a service provided by CSIC, co-funded by iMagine

Terms of use Privacy policy

Login - Register

Marketplace

Search modules

Modules Tools

Train an image classifier
Train your own image classifier with your custom dataset. It comes also pretrained on the 1K ImageNet classes.
Trainable Inference Pre-trained

Object Detection and Classification with Pytorch
A trained Region Convolutional Neural Network (Faster RCNN) for object detection and classification.
Trainable Inference Pre-trained

Object detection
Object detection using (fasterrcnn_pytorch_ap
Trainable Inference

marine_species_seg
WIP Identification of marine species from EMSO Azores deep-sea obervatory

Phytoplankton species classifier (VLIZ)
Identify the species level of Plankton

AI4OS Developm
This is a Docker image modules

iMagine

Login - Register

Module

Marketplace / Phytoplankton species classifier

Phytoplankton species classifier

Build status	License	Created
build unstable	Apache 2.0	2019-01-01

Classify phytoplankton images among 60 classes.

Citizen science has become a powerful force for scientific inquiry, providing researchers with access to a vast array of data points while connecting non scientists to the real process of science. This citizen-researcher relationship creates a very interesting synergy, allowing for the creation, execution, and analysis of research projects. With this in mind, a Convolutional Neural Network has been trained to identify phytoplankton in collaboration with the Vlaams Instituut voor de Zee.

This Docker container contains a trained Convolutional Neural network optimized for phytoplankton identification using images. The architecture used is an Xception [1] network using Keras on top of Tensorflow.

The PREDICT method expects an RGB image as input (or the url of an RGB image) and will return a JSON with the top 5 predictions.

As training dataset we have used a collection of images from the Vlaams Instituut voor de Zee which consists of around 650K images from 60 classes of phytoplankton.

This service is based in the Image Classification with Tensorflow model.

Categories

tensorflow docker deep learning trainable inference pre-trained image classification api-v2 vo.imagine-ai.eu

Additional Resources

Get the code

Github Dockerhub

Get the data

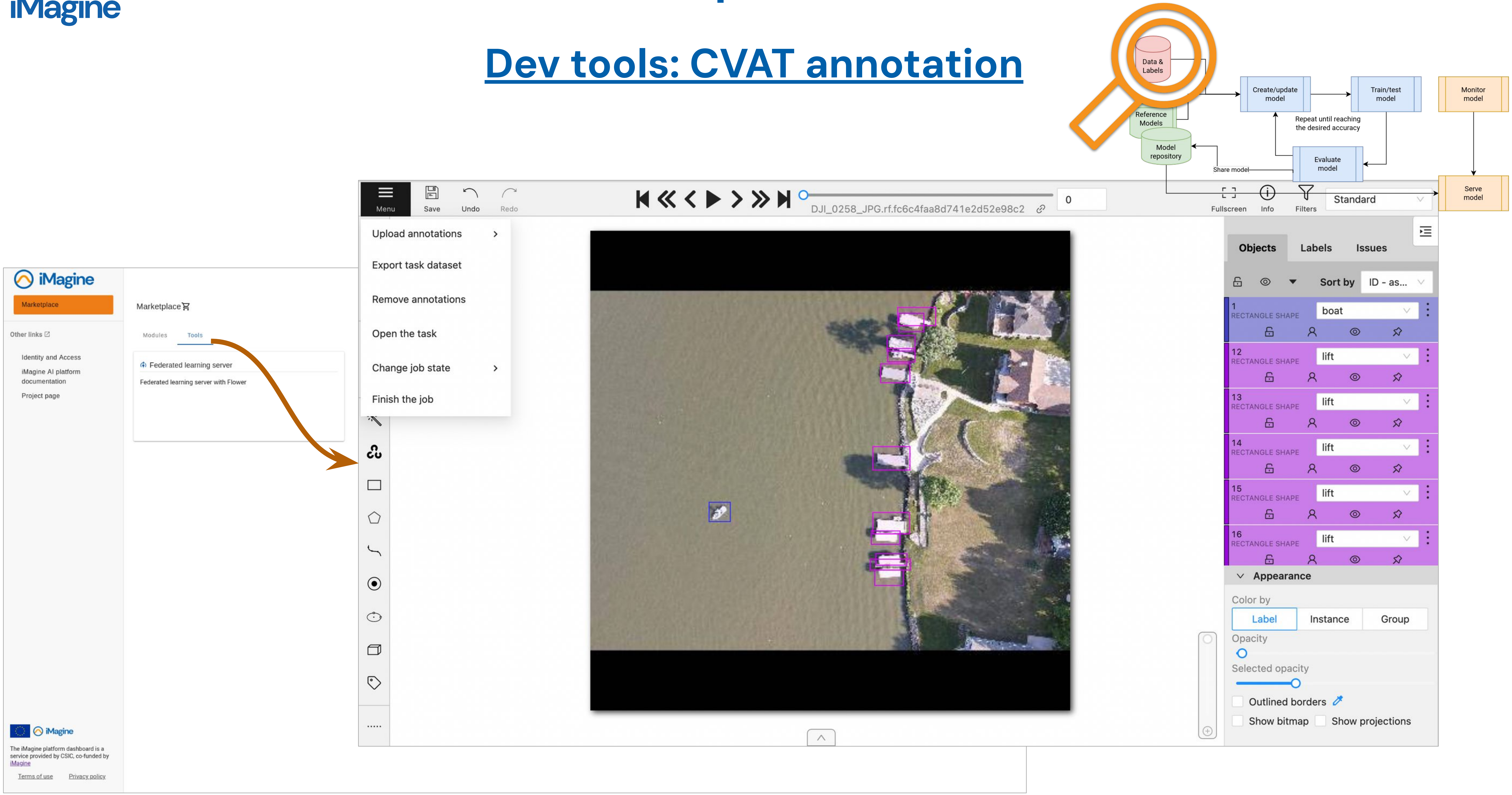
Dataset

https://dashboard.cloud.imagine-ai.eu/marketplace

https://marketplace.eosc-portal.eu/services/imaging-ai-platform-for-aquatic-science

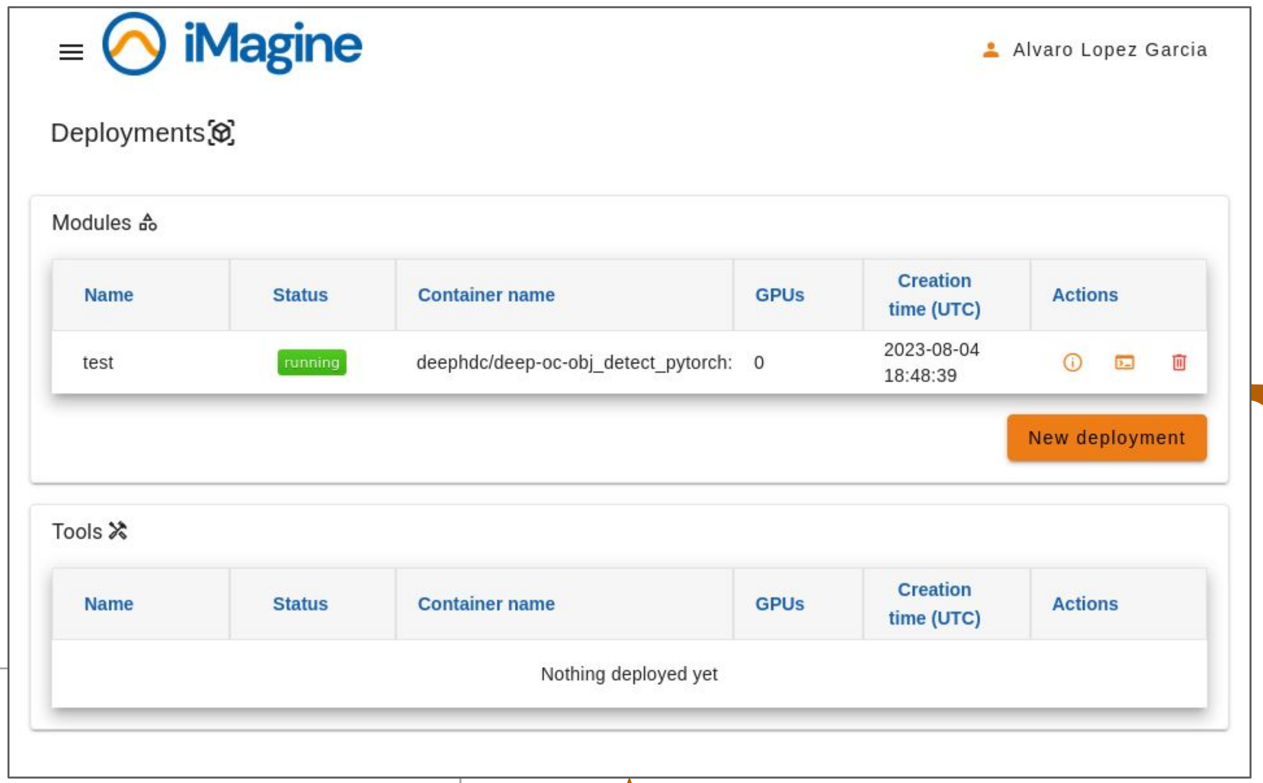
Services for AI/ML development

Dev tools: CVAT annotation



Services for AI/ML development

Dev tools: sandbox and online IDE



iMagine

Alvaro Lopez Garcia

Deployments

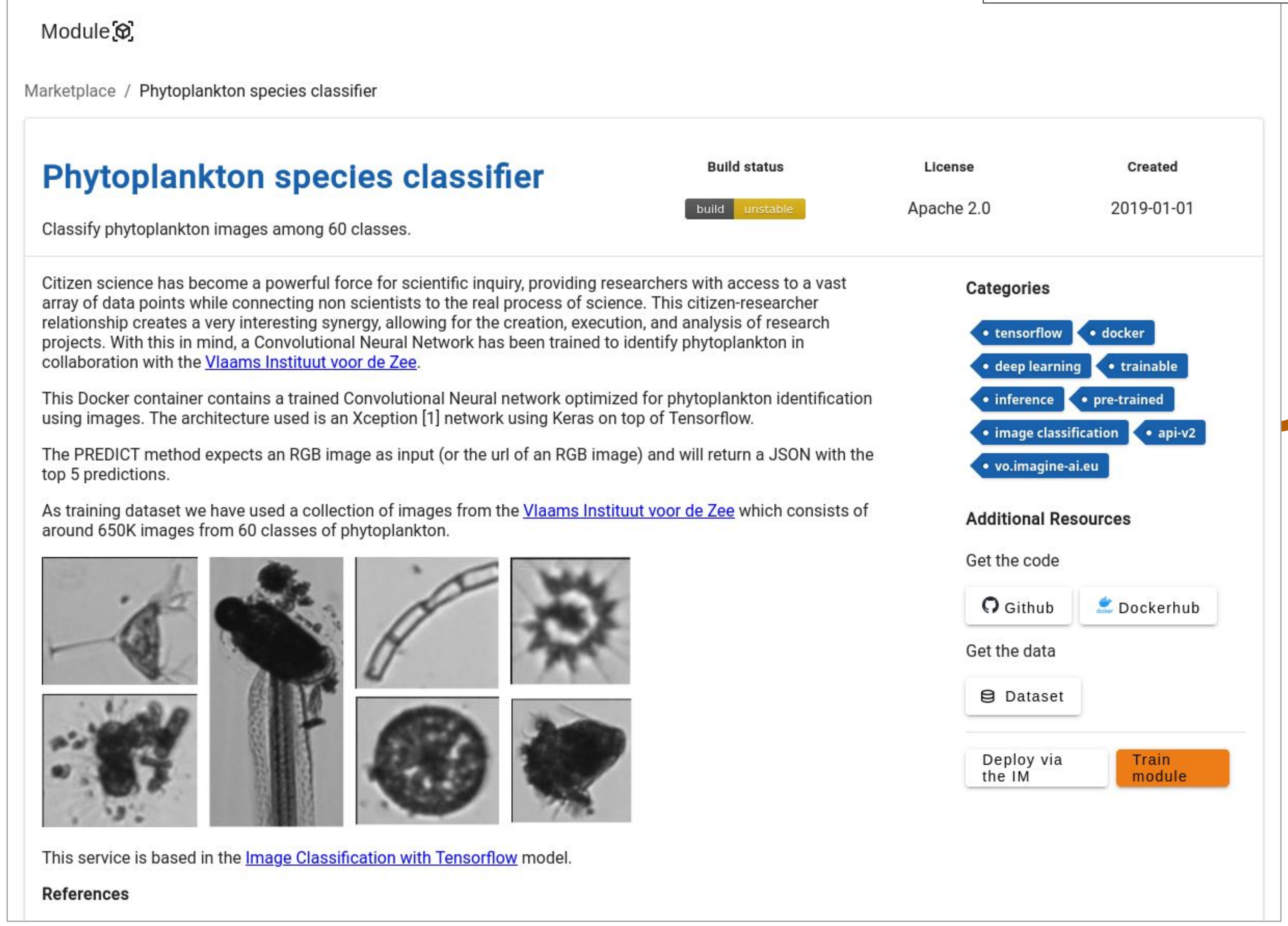
Modules

Name	Status	Container name	GPUs	Creation time (UTC)	Actions
test	running	deephdc/deep-oc-obj_detect_pytorch:	0	2023-08-04 18:48:39	

New deployment

Tools

Name	Status	Container name	GPUs	Creation time (UTC)	Actions
Nothing deployed yet					



iMagine

Module

Marketplace / Phytoplankton species classifier

Phytoplankton species classifier

Build status	License	Created
build unstable	Apache 2.0	2019-01-01

Classify phytoplankton images among 60 classes.

Citizen science has become a powerful force for scientific inquiry, providing researchers with access to a vast array of data points while connecting non scientists to the real process of science. This citizen-researcher relationship creates a very interesting synergy, allowing for the creation, execution, and analysis of research projects. With this in mind, a Convolutional Neural Network has been trained to identify phytoplankton in collaboration with the Vlaams Instituut voor de Zee.

This Docker container contains a trained Convolutional Neural network optimized for phytoplankton identification using images. The architecture used is an Xception [1] network using Keras on top of Tensorflow.

The PREDICT method expects an RGB image as input (or the url of an RGB image) and will return a JSON with the top 5 predictions.

As training dataset we have used a collection of images from the Vlaams Instituut voor de Zee which consists of around 650K images from 60 classes of phytoplankton.

This service is based in the Image Classification with Tensorflow model.

References

Categories

tensorflow, docker, deep learning, trainable, inference, pre-trained, image classification, api-v2, vo.imagine-ai.eu

Additional Resources

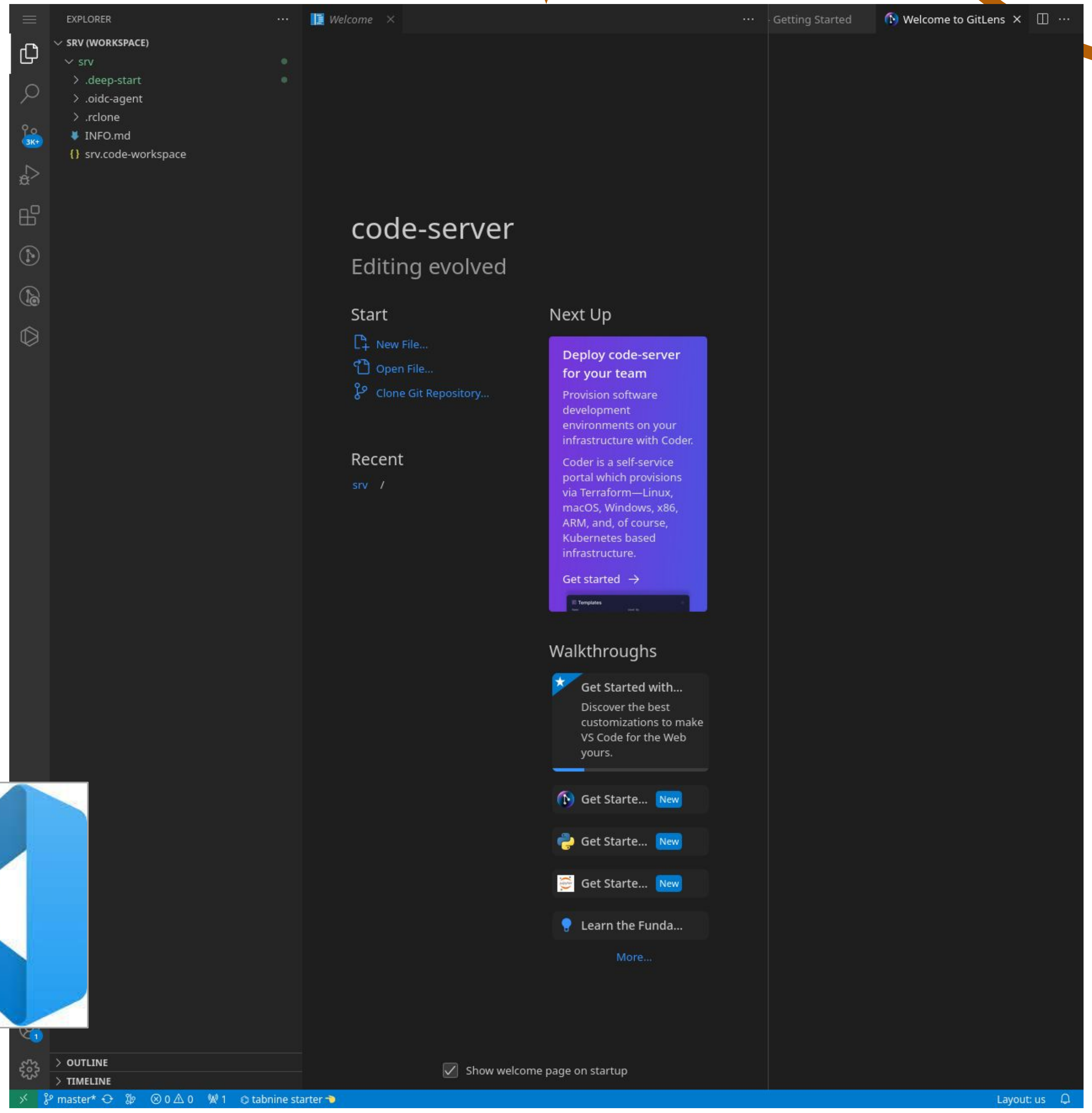
Get the code

Github Dockerhub

Get the data

Dataset

Deploy via the IM | Train module



code-server

Editing evolved

Start

New File...

Open File...

Clone Git Repository...

Recent

srv /

Next Up

Deploy code-server for your team

Provision software development environments on your infrastructure with Coder.

Coder is a self-service portal which provisions via Terraform—Linux, macOS, Windows, x86, ARM, and, of course, Kubernetes based infrastructure.

Get started →

Walkthroughs

Get Started with... Discover the best customizations to make VS Code for the Web yours.

Get Starte... New

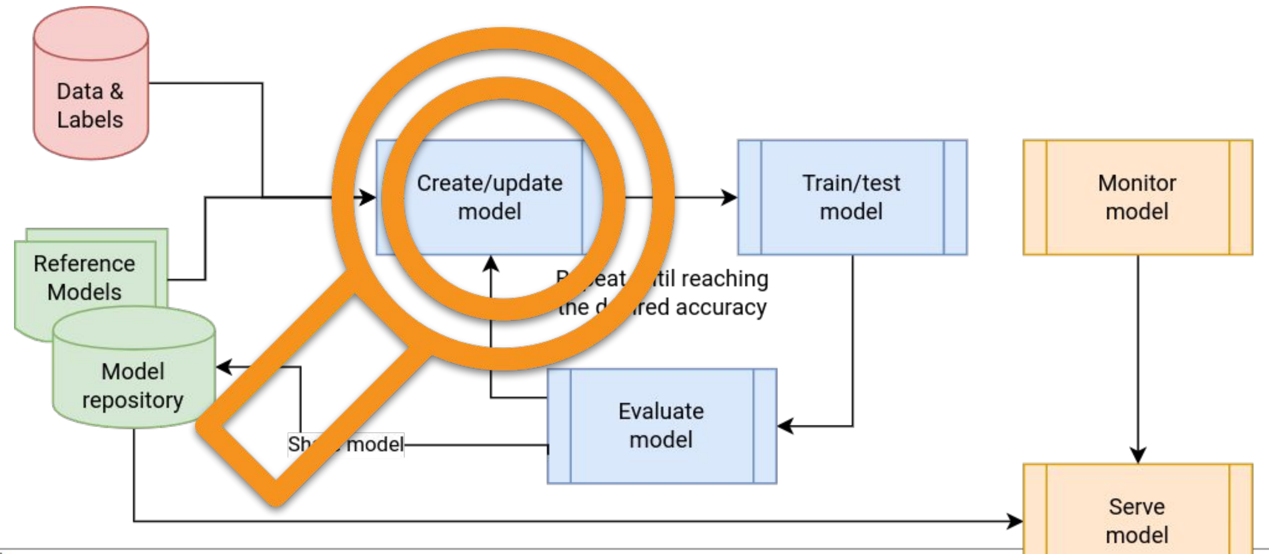
Get Starte... New

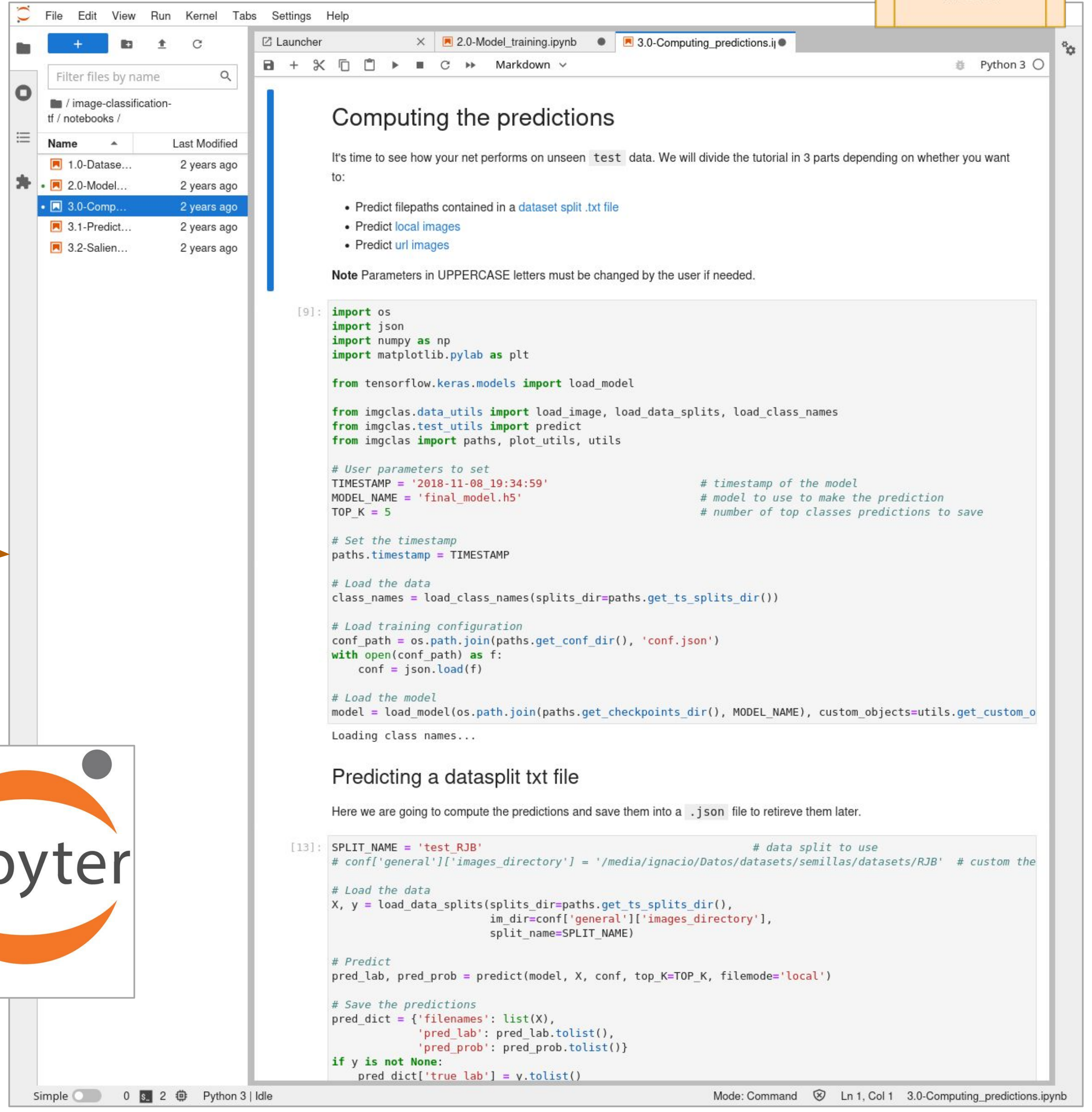
Get Starte... New

Learn the Funda...

More...

☑ Show welcome page on startup





Computing the predictions

It's time to see how your net performs on unseen `test` data. We will divide the tutorial in 3 parts depending on whether you want to:

- Predict filepaths contained in a dataset split .txt file
- Predict local images
- Predict url images

Note Parameters in UPPERCASE letters must be changed by the user if needed.

```
[9]: import os
import json
import numpy as np
import matplotlib.pylab as plt

from tensorflow.keras.models import load_model

from imgclas.data_utils import load_image, load_data_splits, load_class_names
from imgclas.test_utils import predict
from imgclas import paths, plot_utils, utils

# User parameters to set
TIMESTAMP = '2018-11-08_19:34:59'                    # timestamp of the model
MODEL_NAME = 'final_model.h5'                        # model to use to make the prediction
TOP_K = 5                                            # number of top classes predictions to save

# Set the timestamp
paths.timestamp = TIMESTAMP

# Load the data
class_names = load_class_names(splits_dir=paths.get_ts_splits_dir())

# Load training configuration
conf_path = os.path.join(paths.get_conf_dir(), 'conf.json')
with open(conf_path) as f:
    conf = json.load(f)

# Load the model
model = load_model(os.path.join(paths.get_checkpoints_dir(), MODEL_NAME), custom_objects=utils.get_custom_o
```

```
Loading class names...
```

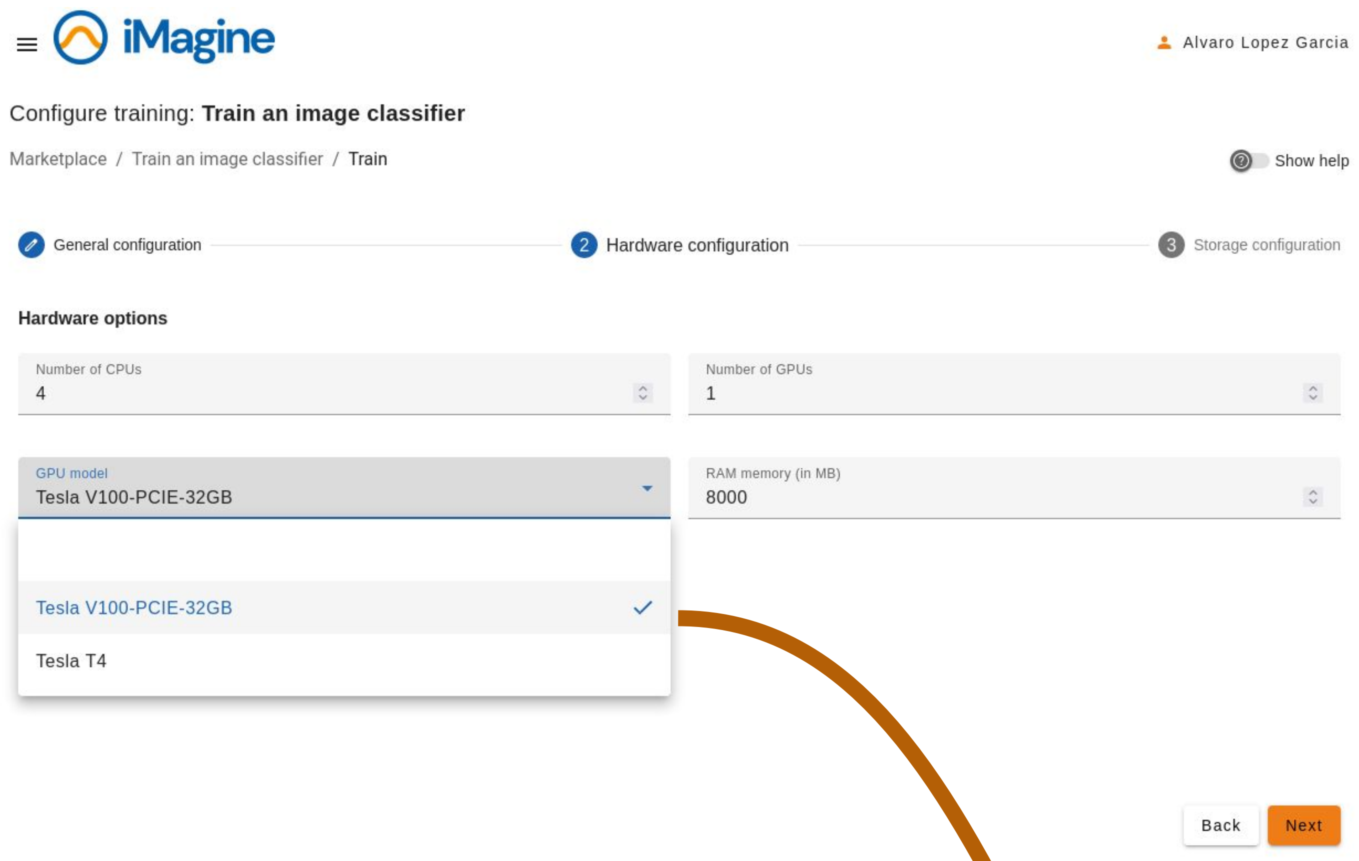
Predicting a datasplit txt file

Here we are going to compute the predictions and save them into a `.json` file to retireve them later.

```
[13]: SPLIT_NAME = 'test_RJB'                                # data split to use
# conf['general']['images_directory'] = '/media/ignacio/Datos/datasets/semillas/datasets/RJB'  # custom the

# Load the data
X, y = load_data_splits(splits_dir=paths.get_ts_splits_dir(),
                        im_dir=conf['general']['images_directory'],
                        split_name=SPLIT_NAME)

# Predict
pred_lab, pred_prob = predict(model, X, conf, top_K=TOP_K, filemode='local')

# Save the predictions
pred_dict = {'filenames': list(X),
             'pred_lab': pred_lab.tolist(),
             'pred_prob': pred_prob.tolist()}
if y is not None:
    pred_dict['true_lab'] = y.tolist()
```

Services for AI/ML development

Training: Transparent GPU access

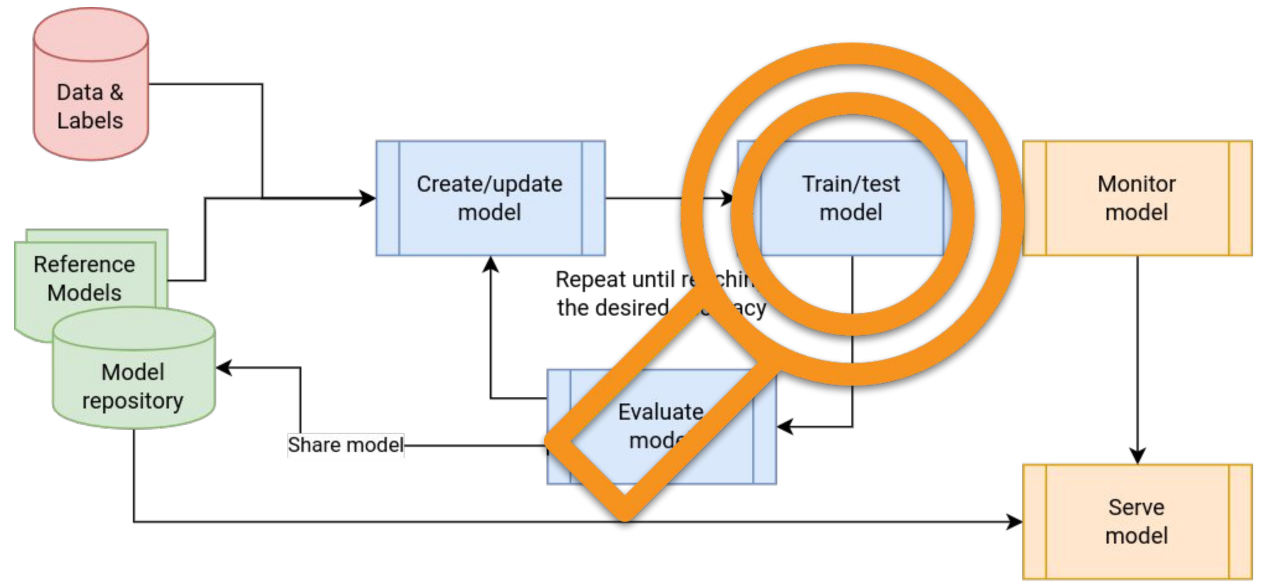


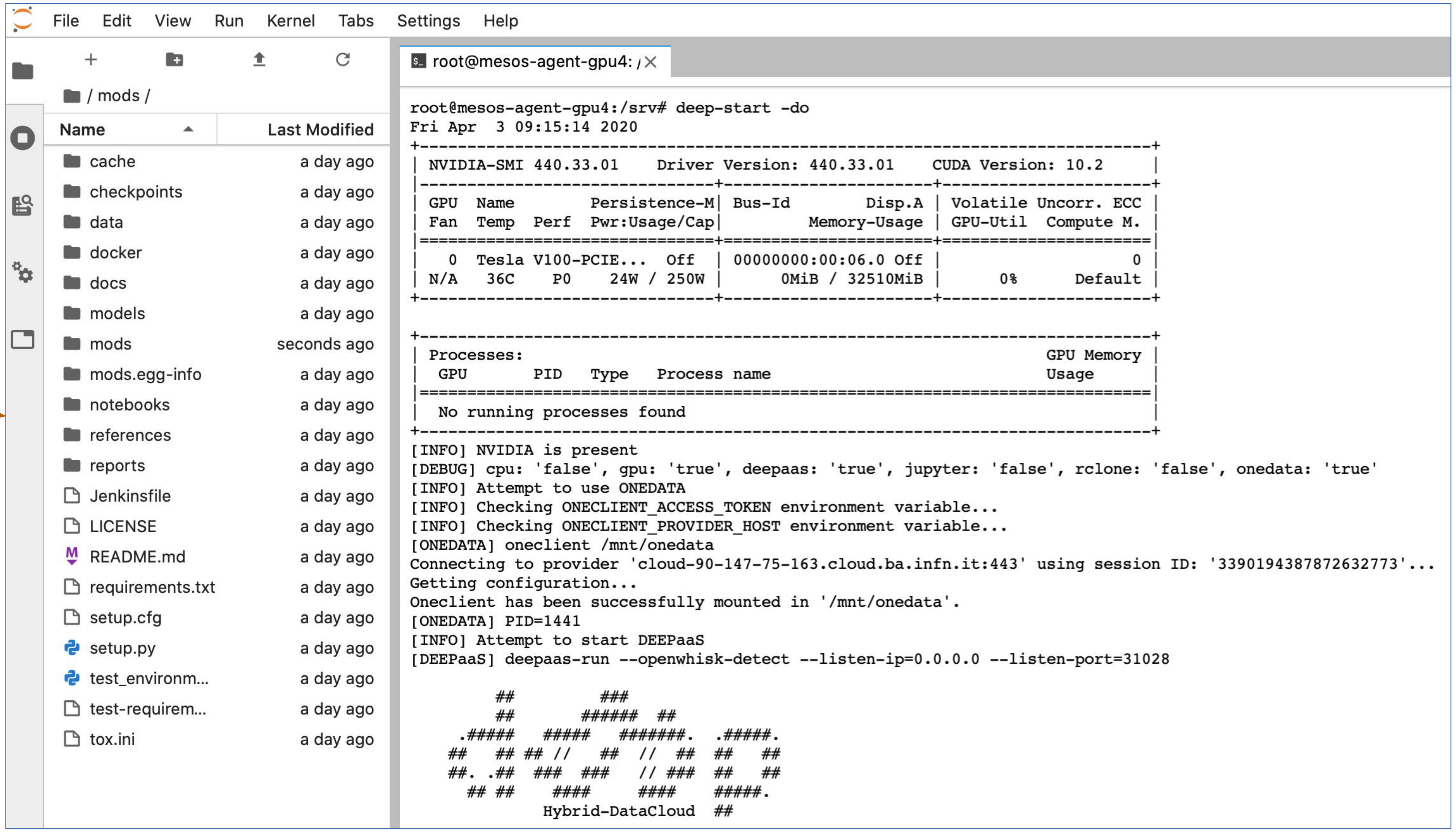
Configure training: **Train an image classifier**

Marketplace / Train an image classifier / Train

Hardware options

Number of CPUs 4	Number of GPUs 1
GPU model Tesla V100-PCIE-32GB	RAM memory (in MB) 8000





/ mods /

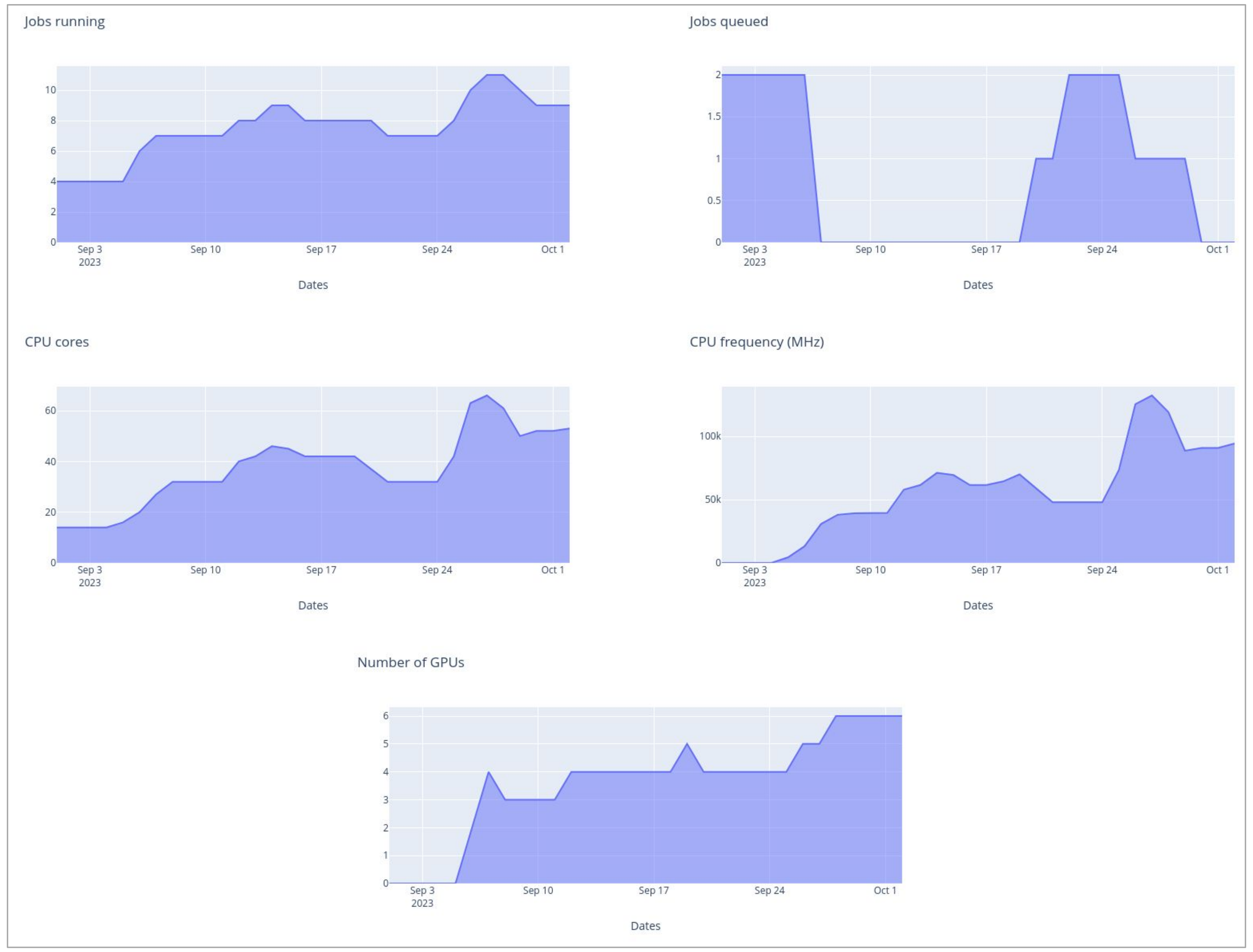
Name	Last Modified
cache	a day ago
checkpoints	a day ago
data	a day ago
docker	a day ago
docs	a day ago
models	a day ago
mods	seconds ago
mods.egg-info	a day ago
notebooks	a day ago
references	a day ago
reports	a day ago
Jenkinsfile	a day ago
LICENSE	a day ago
README.md	a day ago
requirements.txt	a day ago
setup.cfg	a day ago
setup.py	a day ago
test_environm...	a day ago
test-requirem...	a day ago
tox.ini	a day ago

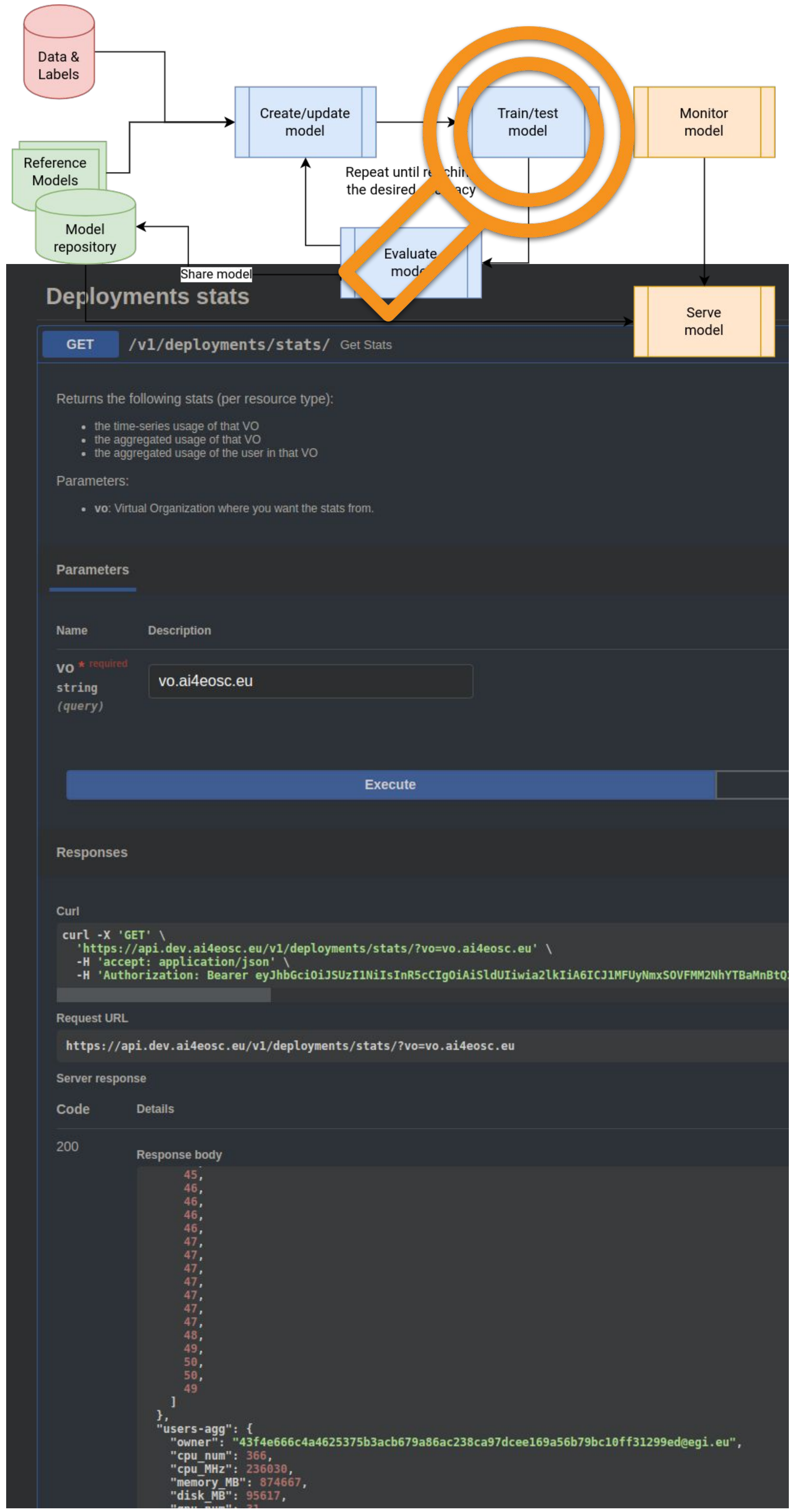
```
root@mesos-agent-gpu4:/srv# deep-start -do
Fri Apr  3 09:15:14 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.33.01    Driver Version: 440.33.01    CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  Off  | 00000000:00:06.0 Off |                    0 |
| N/A   36C    P0    24W / 250W |      0MiB / 32510MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
[INFO] NVIDIA is present
[DEBUG] cpu: 'false', gpu: 'true', deepaas: 'true', jupyter: 'false', rclone: 'false', onedata: 'true'
[INFO] Attempt to use ONEDATA
[INFO] Checking ONECLIENT_ACCESS_TOKEN environment variable...
[INFO] Checking ONECLIENT_PROVIDER_HOST environment variable...
[ONEDATA] oneclient /mnt/onedata
Connecting to provider 'cloud-90-147-75-163.cloud.ba.infn.it:443' using session ID: '3390194387872632773'...
Getting configuration...
Oneclient has been successfully mounted in '/mnt/onedata'.
[ONEDATA] PID=1441
[INFO] Attempt to start DEEPaaS
[DEEPaaS] deepaas-run --openwhisk-detect --listen-ip=0.0.0.0 --listen-port=31028

         ##         ###
         ##       ######  ##
     .#####   #####   #######.  .#####.
    ##   ## ## //   ##  //  ##  ##   ##
    ##. .##  ###  ###   // ###  ##   ##
      ## ##    ####     ####    #####.
              Hybrid-DataCloud  ##
```

Services for AI/ML development

Training: Transparent GPU access

- Automatic platform tasks to collect information from running jobs, then aggregation is performed
- Time series delivered directly by API
- Integration in upstream dashboard is in progress





Deployments stats

GET `/v1/deployments/stats/` Get Stats

Returns the following stats (per resource type):

- the time-series usage of that VO
- the aggregated usage of that VO
- the aggregated usage of the user in that VO

Parameters:

- **vo**: Virtual Organization where you want the stats from.

Parameters

Name	Description
vo * required string (query)	vo.ai4eosc.eu

Execute

Responses

Curl

```
curl -X 'GET' \
  'https://api.dev.ai4eosc.eu/v1/deployments/stats/?vo=vo.ai4eosc.eu' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJ1MFUyNmxSOVFMM2NhYTBaMnBtQ
```

Request URL

```
https://api.dev.ai4eosc.eu/v1/deployments/stats/?vo=vo.ai4eosc.eu
```

Server response

Code Details

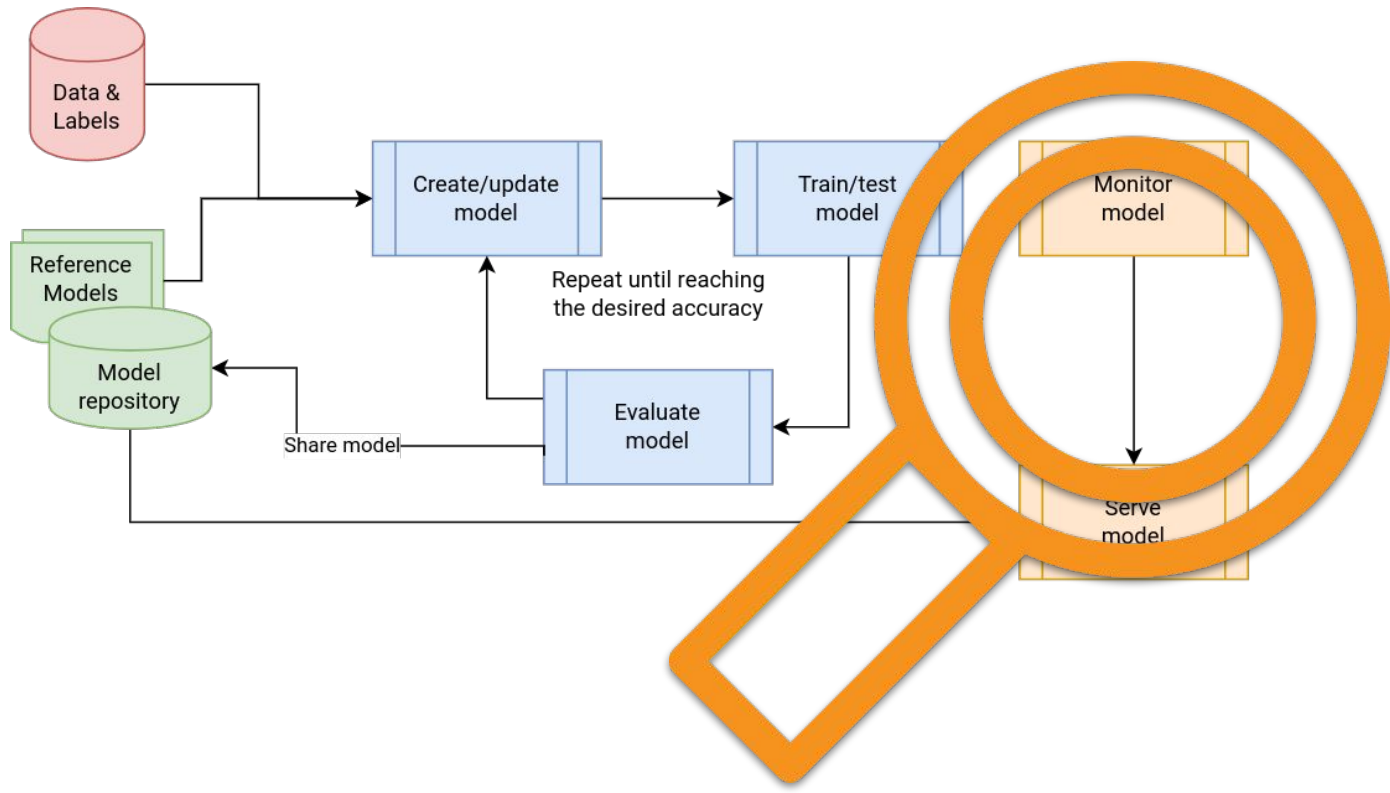
200

Response body

```
        45,
        46,
        46,
        46,
        46,
        47,
        47,
        47,
        47,
        47,
        47,
        47,
        48,
        49,
        50,
        50,
        49
      ]
    },
    "users-agg": {
      "owner": "43f4e666c4a4625375b3acb679a86ac238ca97dcee169a56b79bc10ff31299ed@egi.eu",
      "cpu_num": 366,
      "cpu_MHz": 236030,
      "memory_MB": 874667,
      "disk_MB": 95617,
```

Services for AI/ML development

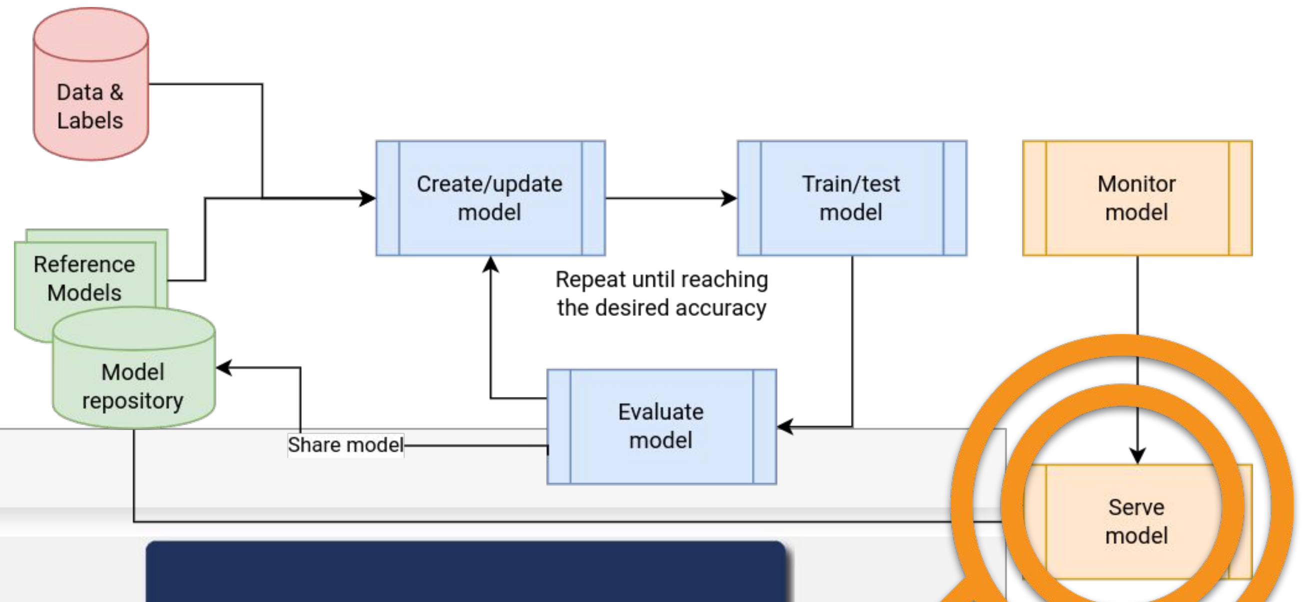
Deploy and monitor

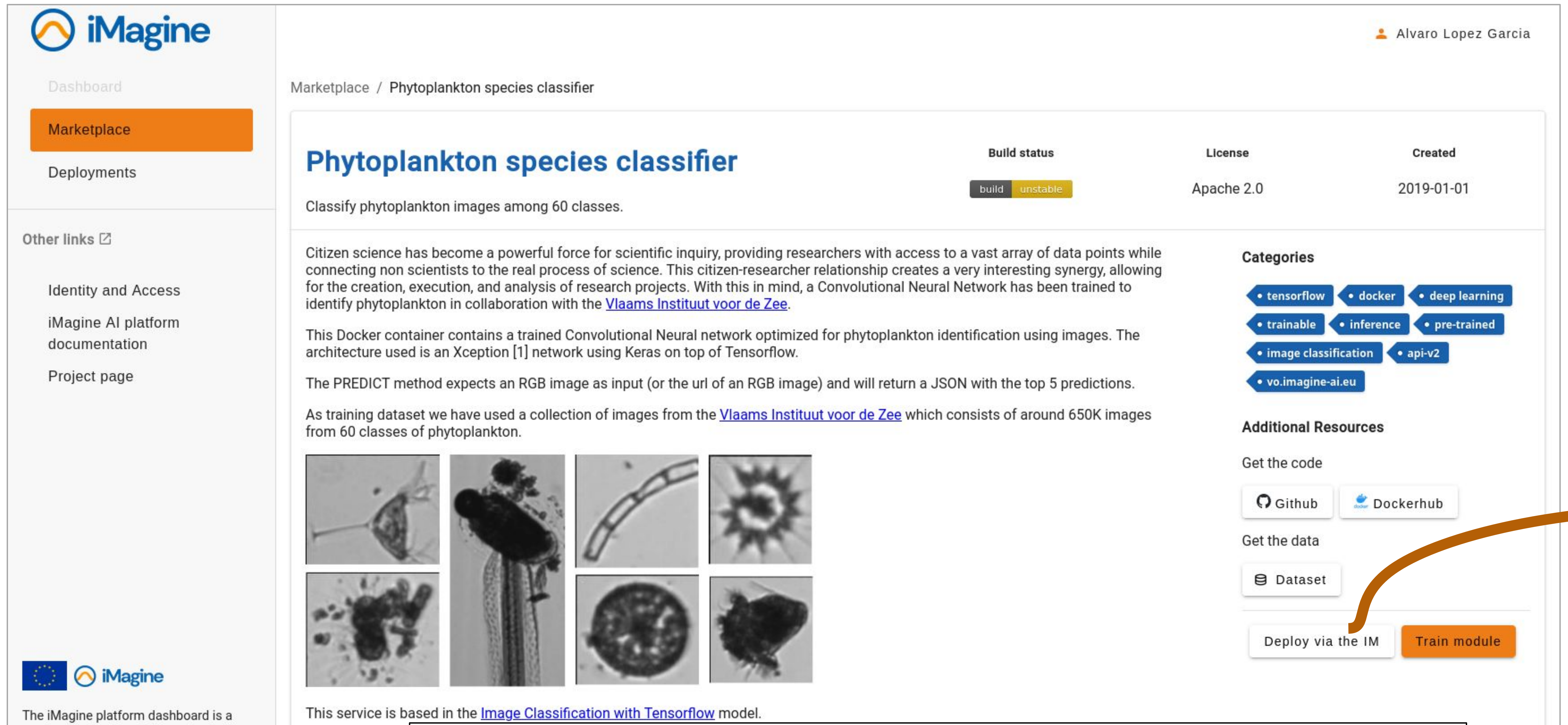


- Deployment
  - Providing services for deployment as services:
    - Through IM an a different cloud (done)
    - Through OSCAR in a platform-managed cluster (partially done)
    - Through OSCAR in user own resources (in progress)
    - Through AI4-PAPI in iMagine AI platform resources (planned)
  - Composite-AI tools (visually build more complex models) (in progress)
- MLOps
  - CI/CD for ML development, deployment, monitoring and operations
- Monitoring
  - Tools to instrument ML models in production (i.e. drift detection)

Services for AI/ML development

Deployment: standalone service





Other links

Identity and Access

iMagine AI platform documentation

Project page

Phytoplankton species classifier

Build status	License	Created
build unstable	Apache 2.0	2019-01-01

Classify phytoplankton images among 60 classes.

Citizen science has become a powerful force for scientific inquiry, providing researchers with access to a vast array of data points while connecting non scientists to the real process of science. This citizen-researcher relationship creates a very interesting synergy, allowing for the creation, execution, and analysis of research projects. With this in mind, a Convolutional Neural Network has been trained to identify phytoplankton in collaboration with the Vlaams Instituut voor de Zee.

This Docker container contains a trained Convolutional Neural network optimized for phytoplankton identification using images. The architecture used is an Xception [1] network using Keras on top of Tensorflow.

The PREDICT method expects an RGB image as input (or the url of an RGB image) and will return a JSON with the top 5 predictions.

As training dataset we have used a collection of images from the Vlaams Instituut voor de Zee which consists of around 650K images from 60 classes of phytoplankton.

This service is based in the Image Classification with Tensorflow model.

References

[1]: Chollet, Fran

Categories

- tensorflow
- docker
- deep learning
- trainable
- inference
- pre-trained
- image classification
- api-v2
- vo.imagine-ai.eu

Additional Resources

Get the code

Github | Dockerhub

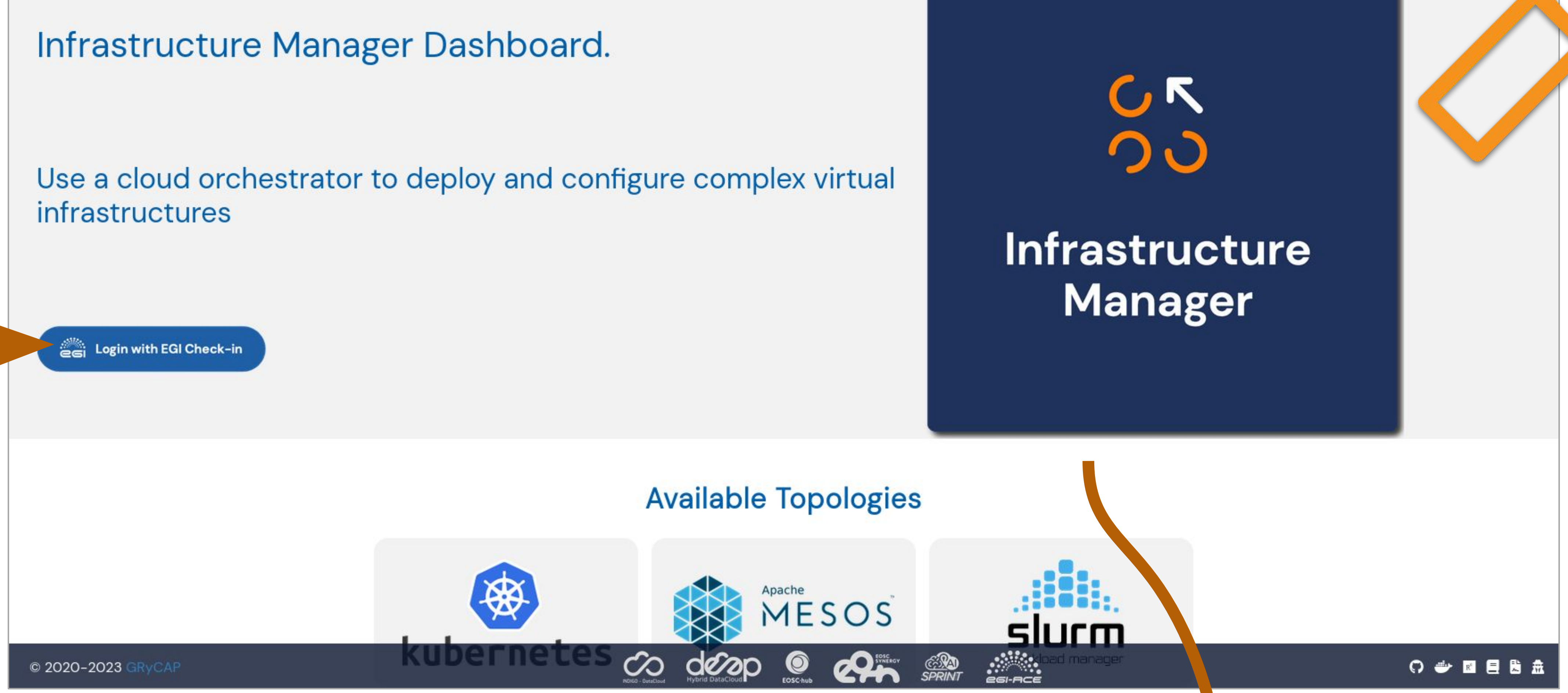
Get the data

Dataset

Deploy via the IM | Train module

The iMagine platform dashboard is a service provided by CSIC, co-funded by iMagine

Terms of use | Privacy policy



Infrastructure Manager Dashboard.

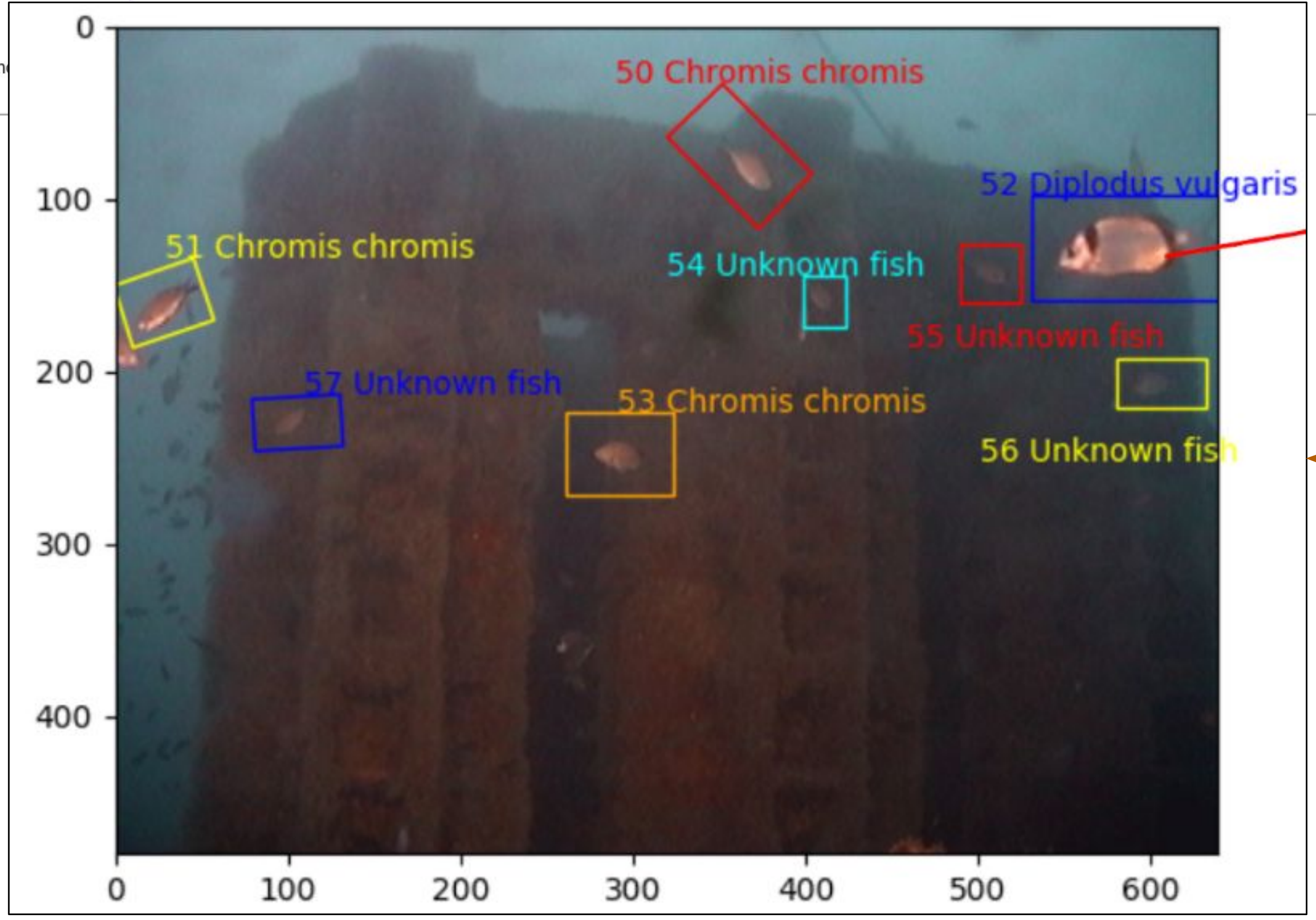
Use a cloud orchestrator to deploy and configure complex virtual infrastructures

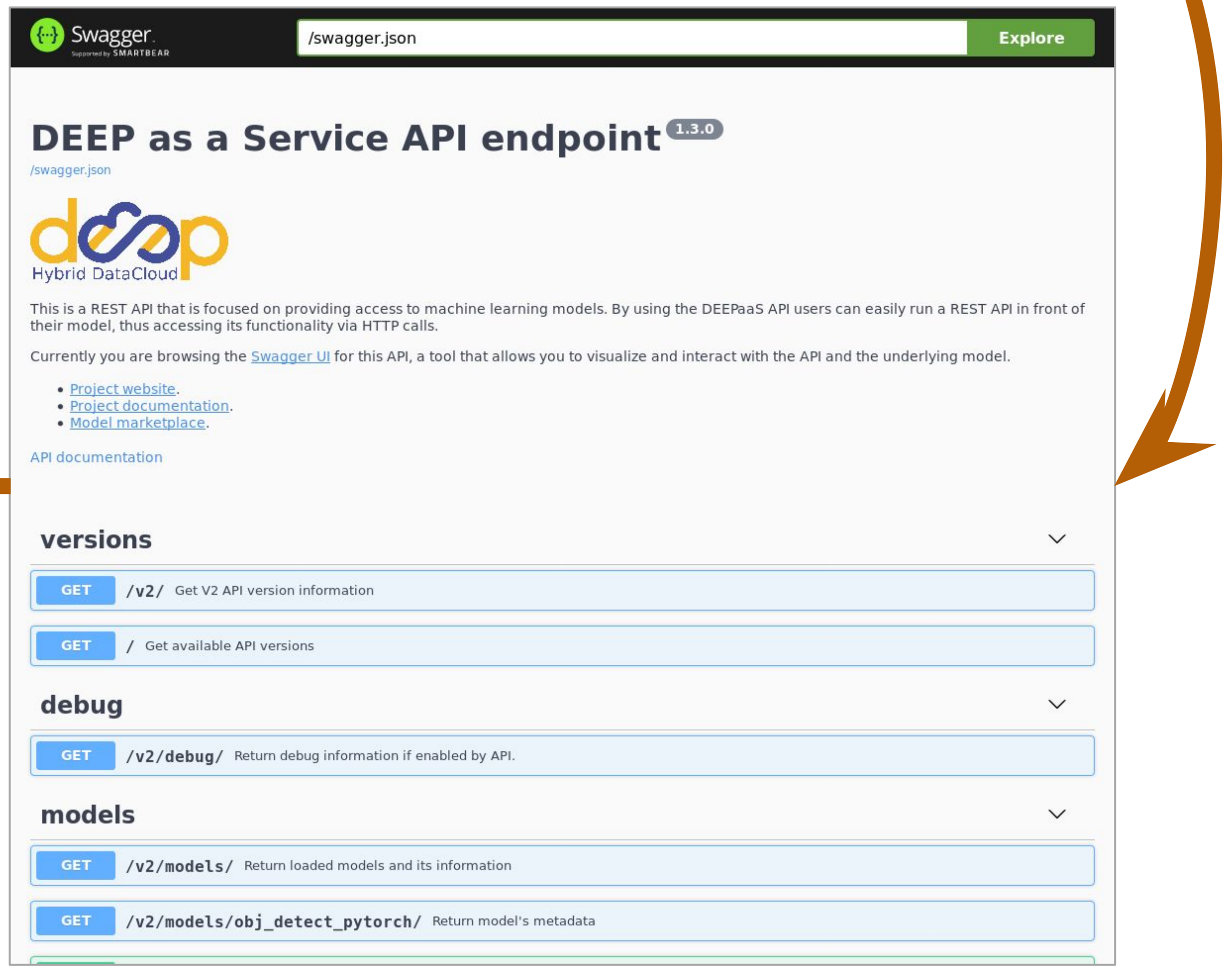
Login with EGI Check-in

Infrastructure Manager

Available Topologies

© 2020–2023 GRyCAP





DEEP as a Service API endpoint 1.3.0

/swagger.json

This is a REST API that is focused on providing access to machine learning models. By using the DEEPaaS API users can easily run a REST API in front of their model, thus accessing its functionality via HTTP calls.

Currently you are browsing the Swagger UI for this API, a tool that allows you to visualize and interact with the API and the underlying model.

- Project website.
- Project documentation.
- Model marketplace.

API documentation

versions

GET /v2/ Get V2 API version information

GET / Get available API versions

debug

GET /v2/debug/ Return debug information if enabled by API.

models

GET /v2/models/ Return loaded models and its information

GET /v2/models/obj_detect_pytorch/ Return model's metadata

Services for AI/ML development

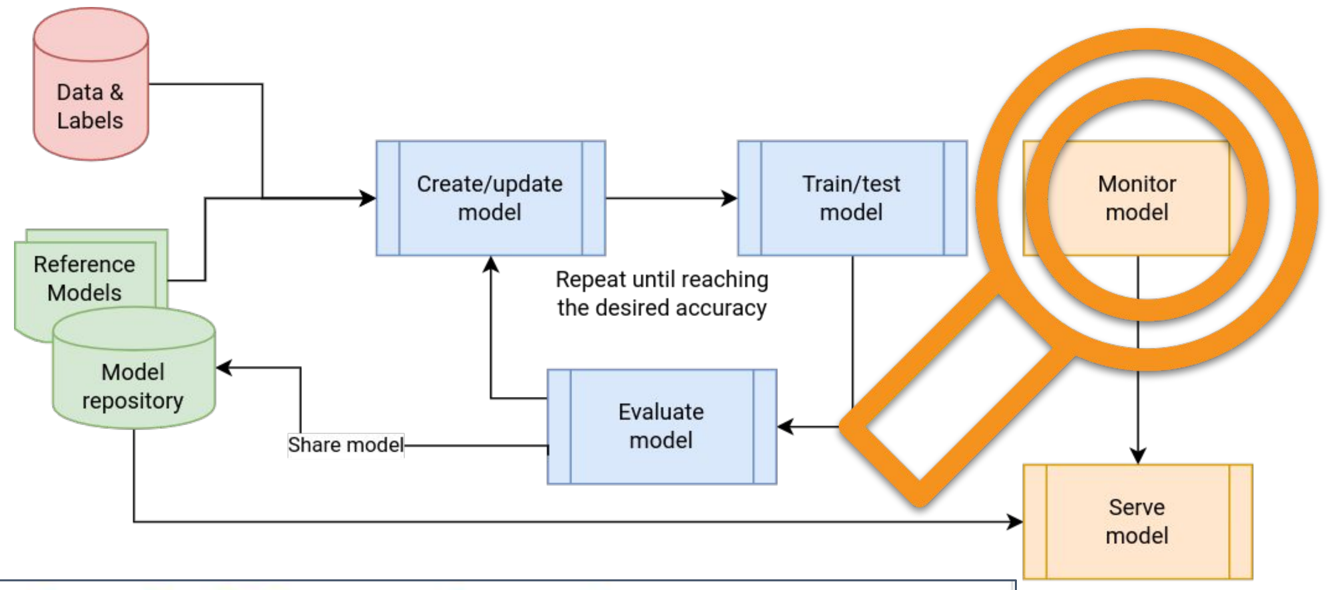
Deployment: Composite AI

Load video
Extract relevant frames
Species identifier 1
Species identifier 2
OSCAR
AI4 eosc
Node-RED
Mean Probabilities
Visualization Result
Data & Labels
Create/update model
Train/test model
Monitor model
Reference Models
Repeat until reaching the desired accuracy
Model repository
Evaluate model
Share model
Serve model

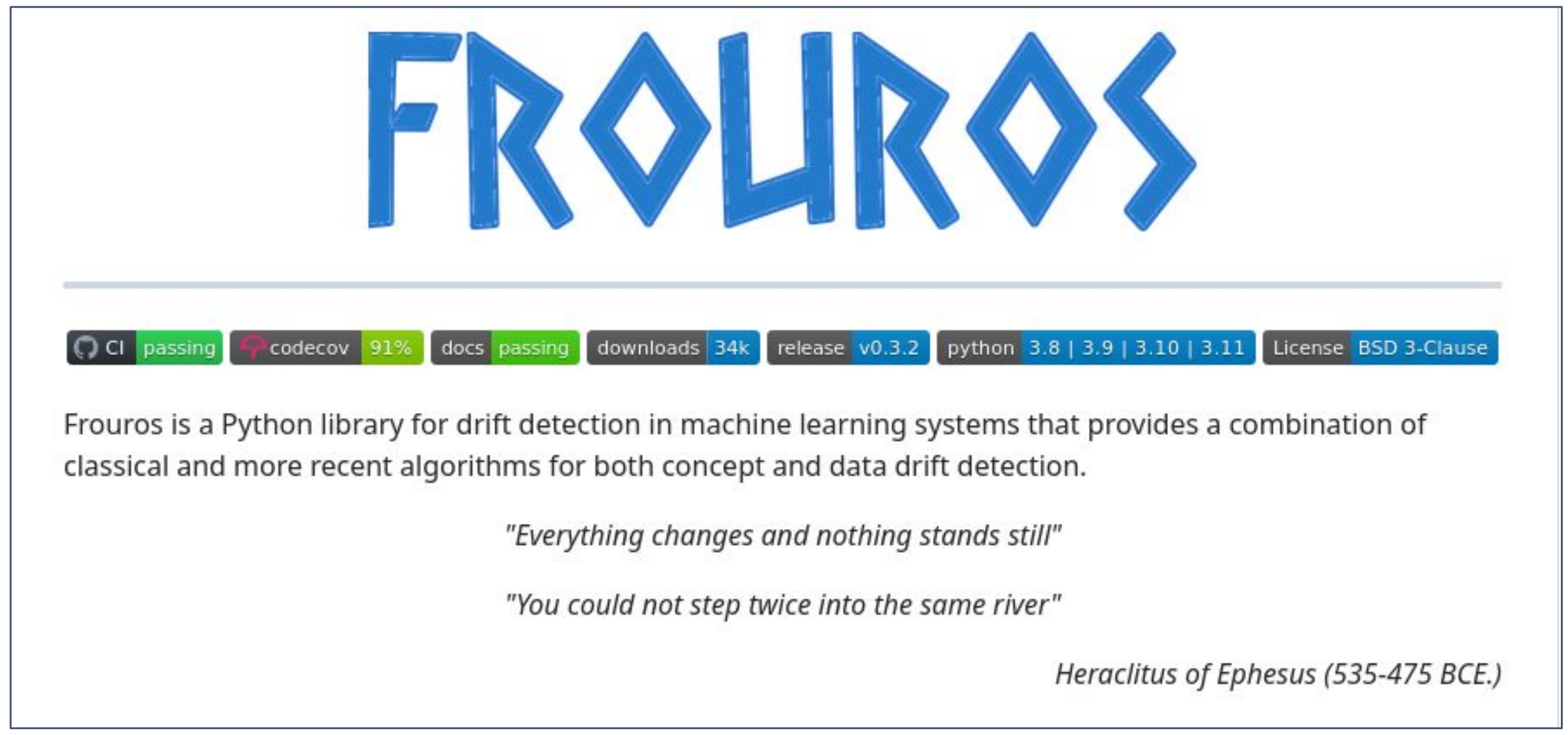
- Use case: multiple AI models can be triggered for inference and later aggregate the results for enhanced accuracy
- Reuse functions (subflow)
- Visual support (drag & drop + customization)
- Minimize orchestration costs

Services for AI/ML development

Deployment: Drift detection



- Monitoring of models in production is not enough
  - Model learns from data, data is not stationary
  - Concept learnt by them model may change over time
- Data and concept drift detection → essential to build more robust models
- Frouros: state-of-the-art library for drift detection in ML problems
  - https://github.com/IFCA/frouros
- Ongoing work towards online services for drift detection → MLOps pipelines with drift detection



FROUROS

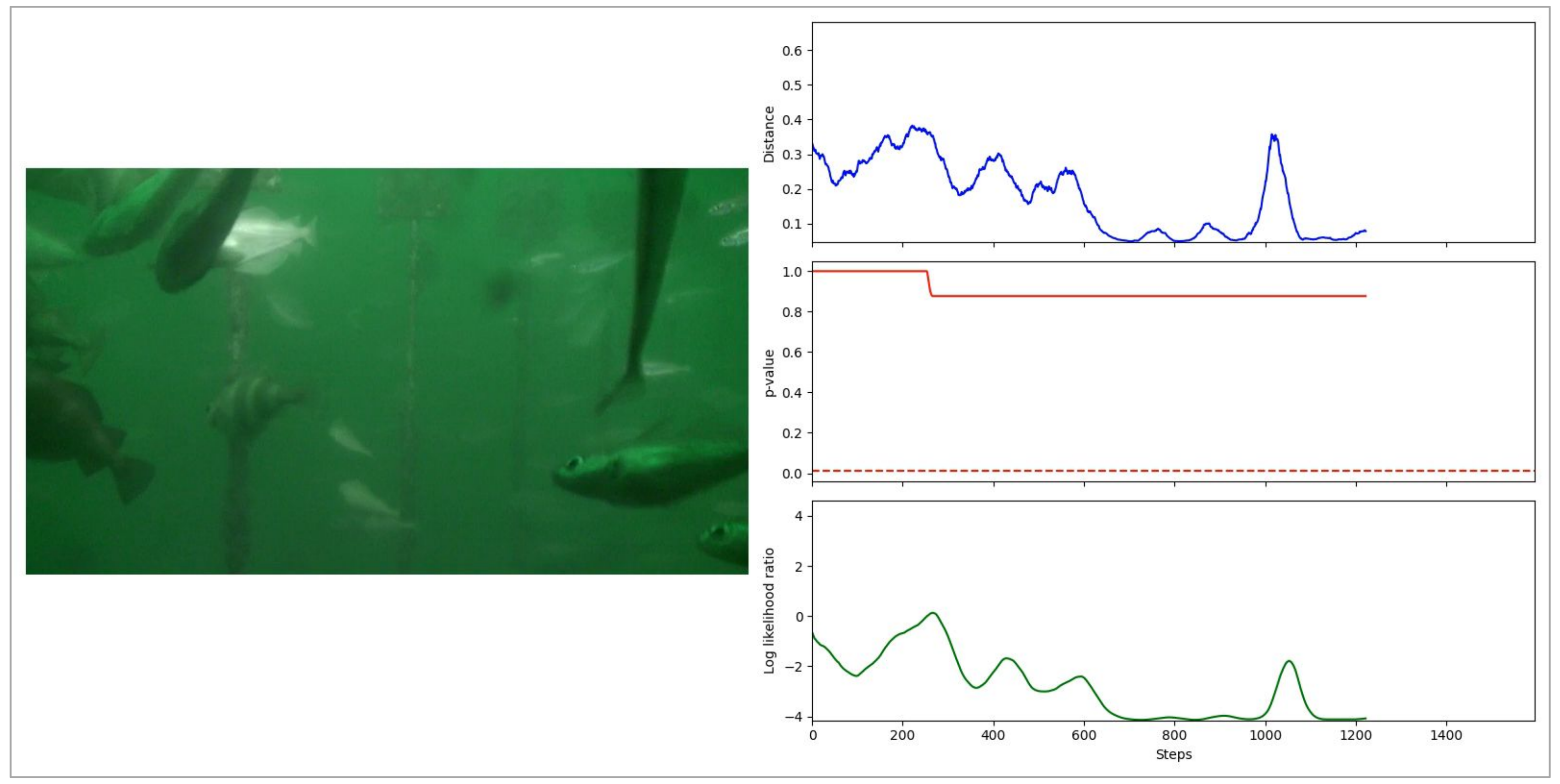
CI passing | codecov 91% | docs passing | downloads 34k | release v0.3.2 | python 3.8 | 3.9 | 3.10 | 3.11 | License BSD 3-Clause

Frouros is a Python library for drift detection in machine learning systems that provides a combination of classical and more recent algorithms for both concept and data drift detection.

"Everything changes and nothing stands still"

"You could not step twice into the same river"

Heraclitus of Ephesus (535-475 BCE.)



Example: data drift detection in underwater video

iMagine

Conclusions, next steps

Wrap-up

Next features

Expected (user) features for 2024

- Final delivery of resource consumption accounting in the dashboard
  - Improving GPU usage and release
  - Work in progress for preemptible, interactive jobs
- Streamlined model deployment as services
  - Including initial MLOps pipelines
- Distributed learning schemes
  - Horovod and Tensorflow parameter server
- Experiment centric dashboard (Q2 2024)
  - Integration of related tools (CVAT), tracking (MLFlow), training deployments and service deployments in a single space
- Integrating metadata schemes in the model registry and dashboard
  - Both generic for ML and domain-specific
  - FAIRness of models and ML assets

iMagine

Thank you!

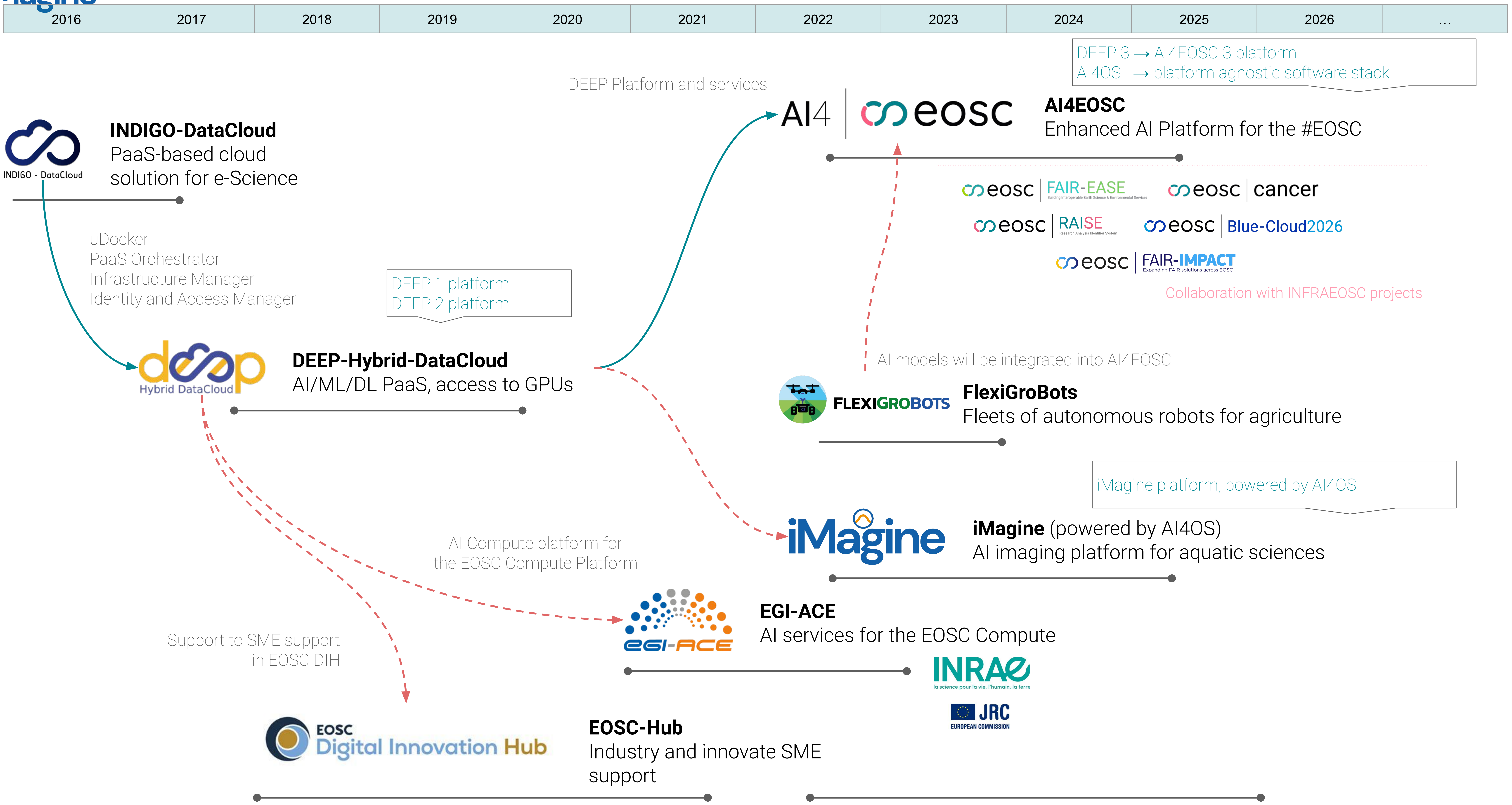
imagine-ai.eu

IiMagine receives funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101058625

iMagine

Backup slides

Background, ecosystem, collaborations



DEEP 3 ⟶ AI4EOSC 3 platform
AI4OS ⟶ platform agnostic software stack

DEEP Platform and services

AI4EOSC
Enhanced AI Platform for the #EOSC

Collaboration with INFRAEOSC projects

INDIGO-DataCloud
PaaS-based cloud solution for e-Science

uDocker
PaaS Orchestrator
Infrastructure Manager
Identity and Access Manager

DEEP 1 platform
DEEP 2 platform

DEEP-Hybrid-DataCloud
AI/ML/DL PaaS, access to GPUs

AI models will be integrated into AI4EOSC

FlexiGroBots
Fleets of autonomous robots for agriculture

iMagine platform, powered by AI4OS

iMagine (powered by AI4OS)
AI imaging platform for aquatic sciences

AI Compute platform for the EOSC Compute Platform

EGI-ACE
AI services for the EOSC Compute

Support to SME support in EOSC DIH

EOSC-Hub
Industry and innovate SME support

iMagine

For Operators

Platform features

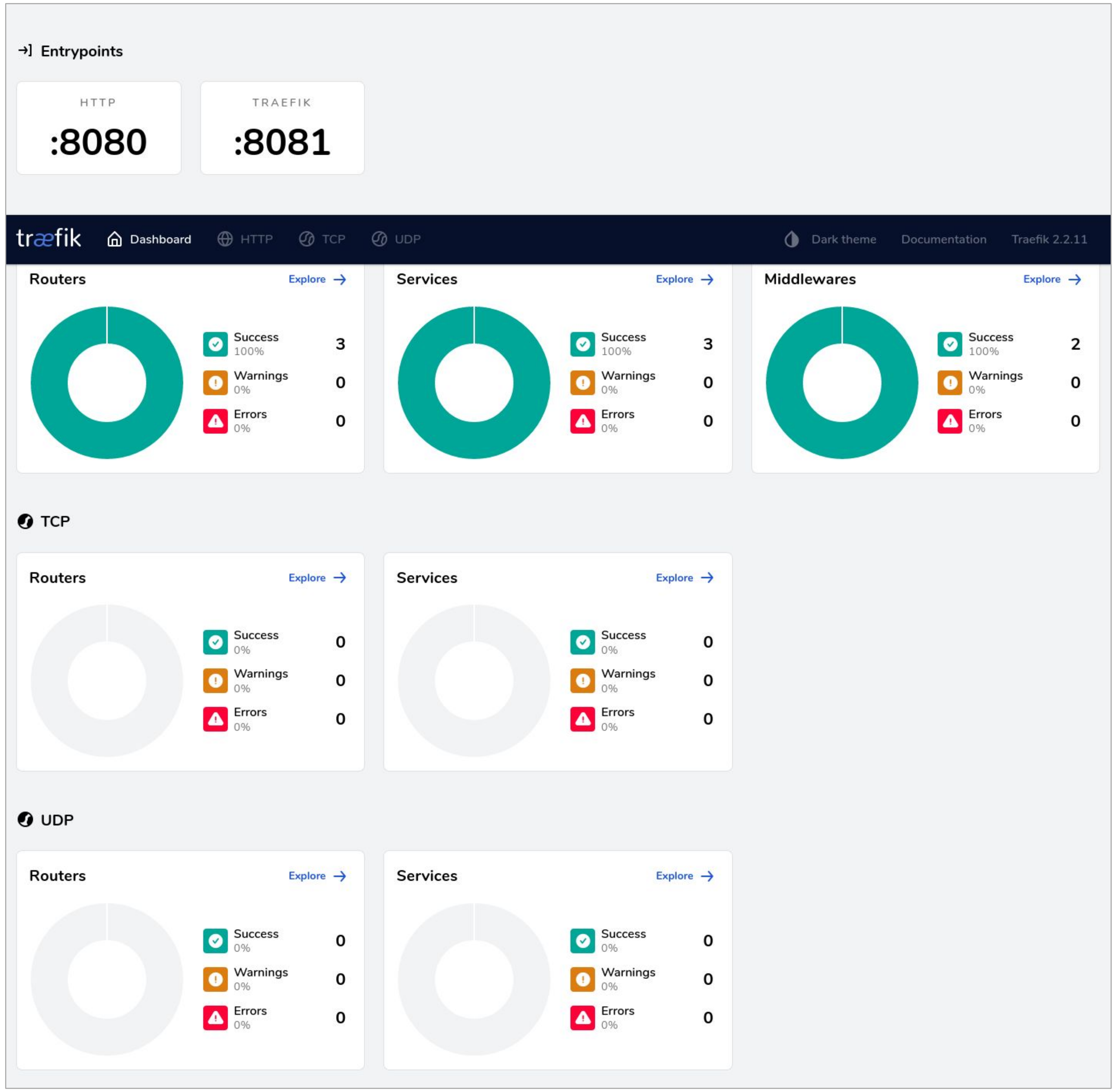
Multi-site deployment

A service mesh approach

- Platform based on service mesh approach
  - Nomad: as workload management system
  - Consul: to enable service mesh
  - Traefik: to provide LB and proxy for running jobs
- AI4-PAPI (AI4 - Platform API)
  - Unified access to the platform, integrated with EGI AAI
  - Share-nothing architecture → horizontal scalability
- Automation of the deployment through Ansible roles
- Additional job (task) types to deliver more complex services
  - E.g. Image annotation, etc.
- AI4OS platform-wide container registry
  - Harbor https://registry.services.ai4os.eu/

Traefik as service proxy

- Providing access to underlying tasks
- Rick monitoring of traffic status
- Multi-traefik deployment
  - 1 per Nomad datacenter
  - Pro: ensure access to user jobs in case of failure of other sites
  - Con: hostname will change if tasks are migrated
- Dynamic creation of endpoints (secure) for user tasks, e.g.:
  - IDE
  - API
  - Monitoring
  - Federated server



Entrypoints

HTTP :8080

TRAEFIK :8081

Routers — Success 100% 3; Warnings 0% 0; Errors 0% 0

Services — Success 100% 3; Warnings 0% 0; Errors 0% 0

Middlewares — Success 100% 2; Warnings 0% 0; Errors 0% 0

TCP

Routers — Success 0% 0; Warnings 0% 0; Errors 0% 0

Services — Success 0% 0; Warnings 0% 0; Errors 0% 0

UDP

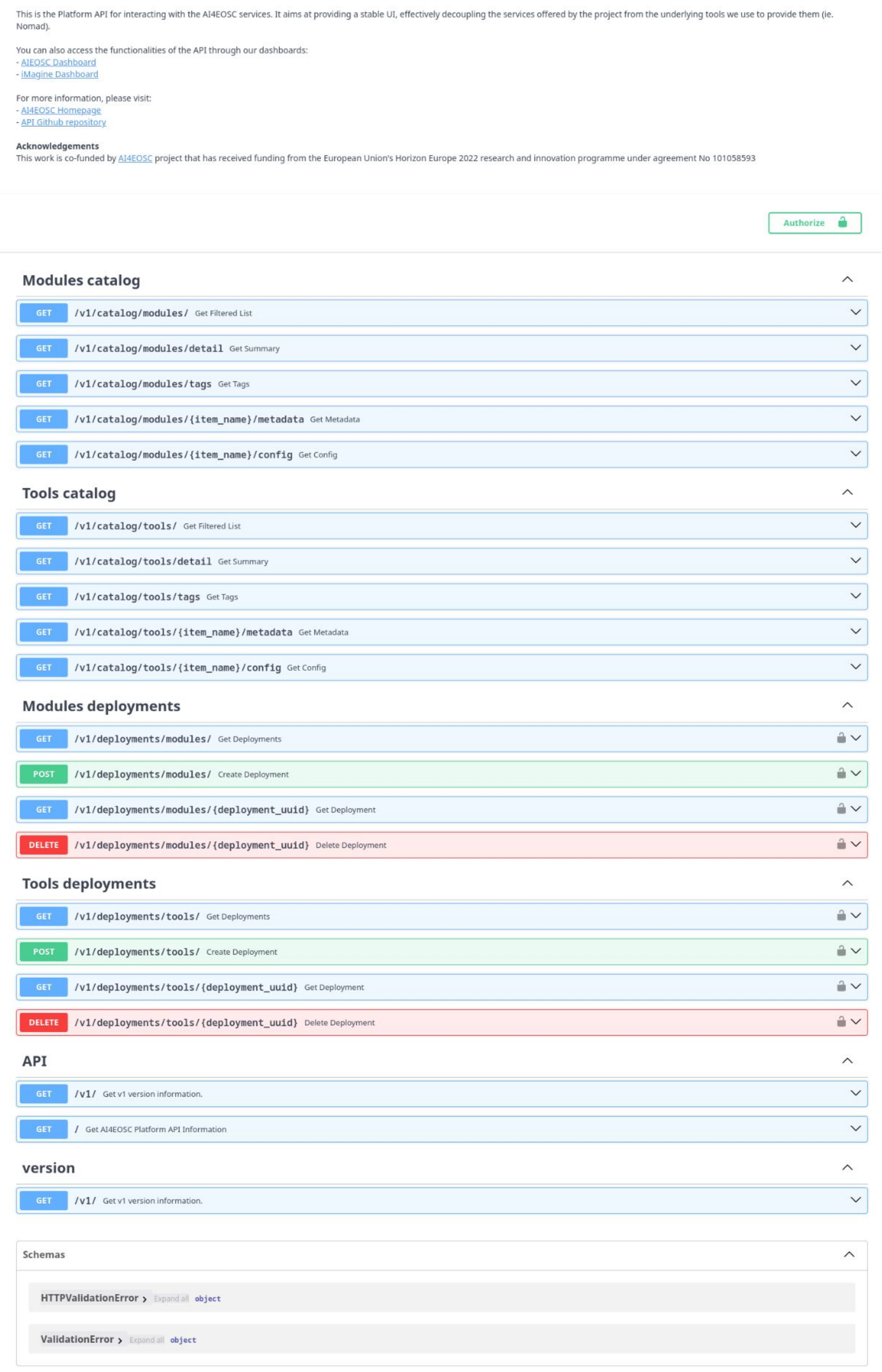
Routers — Success 0% 0; Warnings 0% 0; Errors 0% 0

Services — Success 0% 0; Warnings 0% 0; Errors 0% 0

AI4-Platform API

Operator features

- Nomad provides very basic AuthN/Z system
- AI4-PAPI proxies user requests to the platform
  - i.e. no direct access to Nomad for the users
- Additional sidecar containers and tasks
  - Storage sidecar container
    - Currently supporting remote mounting through rclone
    - Integrating with EOSC-RAISE storage
  - Creation of metering tasks
    - Fine-grained accounting system
  - Execution of complex tooling deployments
    - E.g. image annotation, development environments, tracking services…
- Multi-API deployments are possible
  - Share-nothing architecture and stateless service



This is the Platform API for interacting with the AI4EOSC services. It aims at providing a stable UI, effectively decoupling the services offered by the project from the underlying tools we use to provide them (ie. Nomad).

You can also access the functionalities of the API through our dashboards:
- AIEOSC Dashboard
- iMagine Dashboard

For more information, please visit:
- AI4EOSC Homepage
- API Github repository

Acknowledgements
This work is co-funded by AI4EOSC project that has received funding from the European Union's Horizon Europe 2022 research and innovation programme under agreement No 101058593

Modules catalog

GET `/v1/catalog/modules/` Get Filtered List
GET `/v1/catalog/modules/detail` Get Summary
GET `/v1/catalog/modules/tags` Get Tags
GET `/v1/catalog/modules/{item_name}/metadata` Get Metadata
GET `/v1/catalog/modules/{item_name}/config` Get Config

Tools catalog

GET `/v1/catalog/tools/` Get Filtered List
GET `/v1/catalog/tools/detail` Get Summary
GET `/v1/catalog/tools/tags` Get Tags
GET `/v1/catalog/tools/{item_name}/metadata` Get Metadata
GET `/v1/catalog/tools/{item_name}/config` Get Config

Modules deployments

GET `/v1/deployments/modules/` Get Deployments
POST `/v1/deployments/modules/` Create Deployment
GET `/v1/deployments/modules/{deployment_uuid}` Get Deployment
DELETE `/v1/deployments/modules/{deployment_uuid}` Delete Deployment

Tools deployments

GET `/v1/deployments/tools/` Get Deployments
POST `/v1/deployments/tools/` Create Deployment
GET `/v1/deployments/tools/{deployment_uuid}` Get Deployment
DELETE `/v1/deployments/tools/{deployment_uuid}` Delete Deployment

API

GET `/v1/` Get v1 version information.
GET `/` Get AI4EOSC Platform API Information

version

GET `/v1/` Get v1 version information.

Schemas

HTTPValidationError › Expand all object
ValidationError › Expand all object

iMagine

Platform next steps

Expected (operators) features for 2024

- Streamlining integration of new sites
  - Semi-automatically through Ansible (done)
  - Implement automation through IM (mid-term)

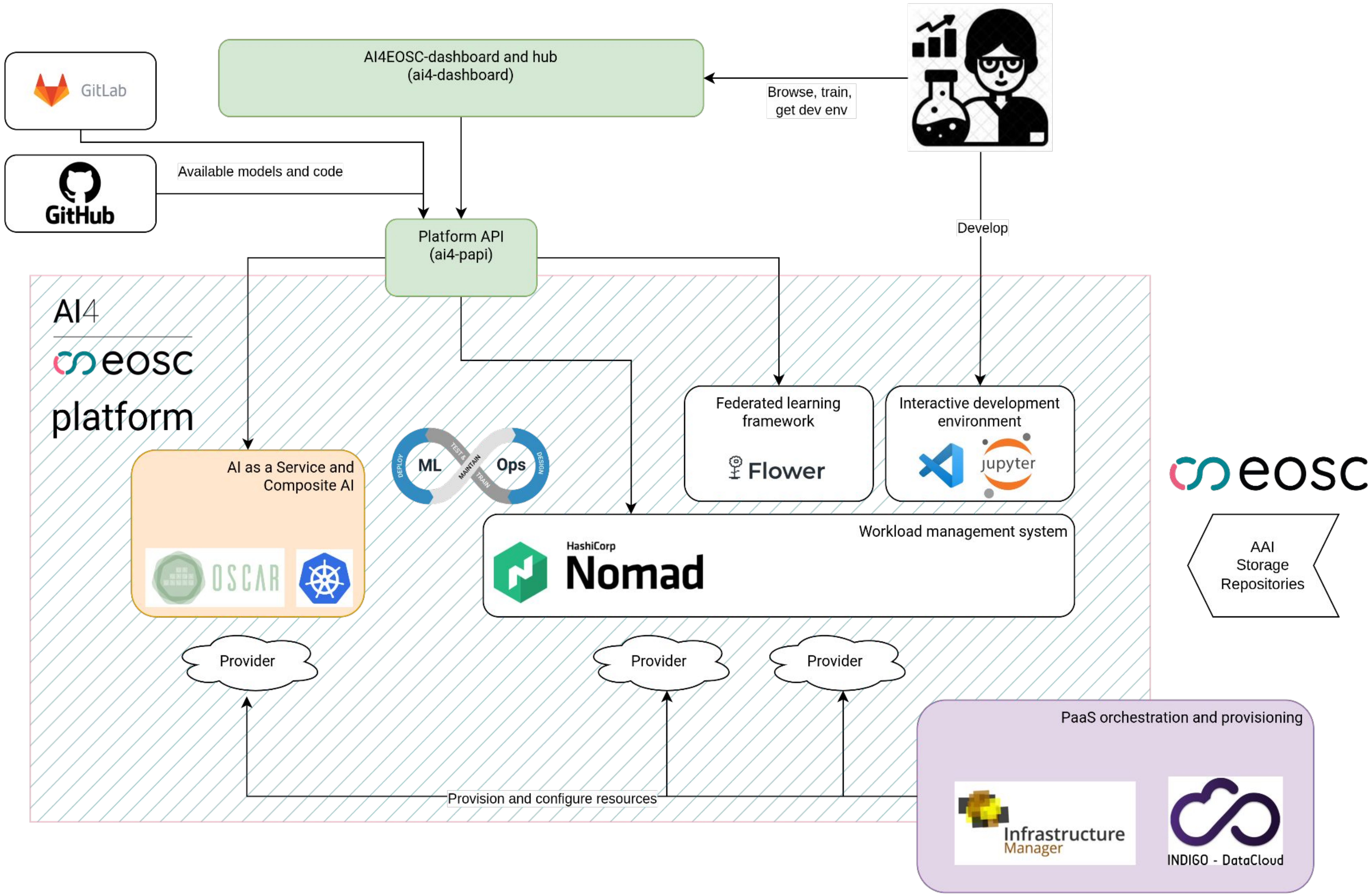
- Delivering iMagine AI Application Deployment Service
  - Initial work ongoing, but waiting for WP5 work to start
  - Different possibilities for deployment:
    - Through IM an a different cloud (done)
    - Through OSCAR in a platform-managed cluster (partially done)
    - Through OSCAR in user own resources (in progress)
    - Through AI4-PAPI in iMagine AI platform resources (planned)

iMagine

For Operators

Platform Architecture

AI4OS high level architecture

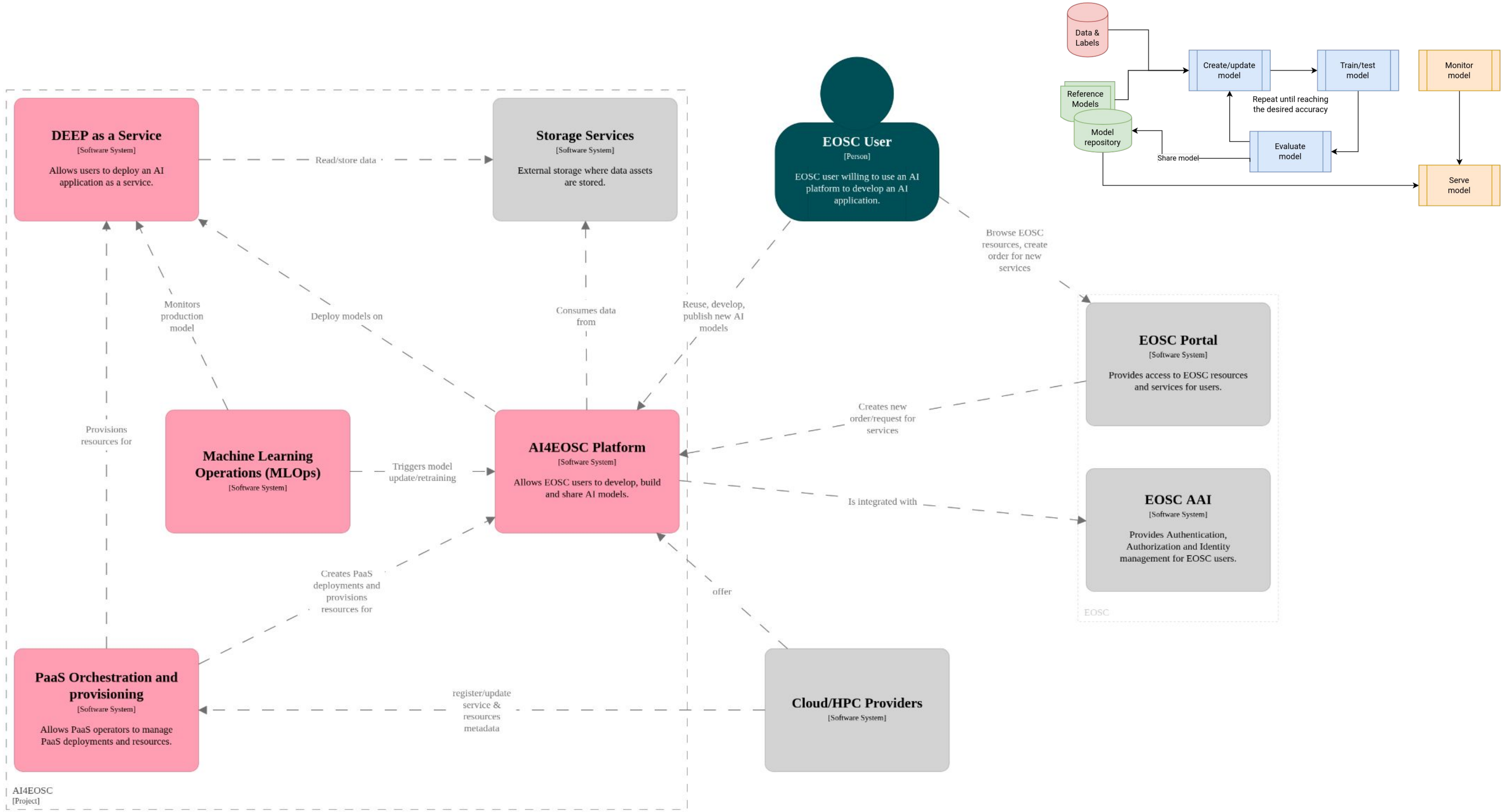


GitLab
GitHub
Available models and code
AI4EOSC-dashboard and hub (ai4-dashboard)
Browse, train, get dev env
Develop
Platform API (ai4-papi)
AI4 eosc platform
AI as a Service and Composite AI
OSCAR
ML Ops
Federated learning framework
Flower
Interactive development environment
jupyter
Workload management system
HashiCorp Nomad
eosc
AAI
Storage
Repositories
Provider
Provider
Provider
Provision and configure resources
PaaS orchestration and provisioning
Infrastructure Manager
INDIGO - DataCloud

Detailed C4 architecture can be found here:

- Workspace https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1
- Diagrams: https://structurizr.com/share/73873/2f769b91-f208-41b0-b79f-5e196435bdb1/images

iMagine AI Platform system context (C4 Model)


Data & Labels
Reference Models
Model repository
Create/update model
Train/test model
Repeat until reaching the desired accuracy
Evaluate model
Share model
Monitor model
Serve model
DEEP as a Service
[Software System]
Allows users to deploy an AI application as a service.
Read/store data
Storage Services
[Software System]
External storage where data assets are stored.
EOSC User
[Person]
EOSC user willing to use an AI platform to develop an AI application.
Browse EOSC resources, create order for new services
Monitors production model
Deploy models on
Consumes data from
Reuse, develop, publish new AI models
EOSC Portal
[Software System]
Provides access to EOSC resources and services for users.
Provisions resources for
Machine Learning Operations (MLOps)
[Software System]
Triggers model update/retraining
AI4EOSC Platform
[Software System]
Allows EOSC users to develop, build and share AI models.
Creates new order/request for services
Is integrated with
EOSC AAI
[Software System]
Provides Authentication, Authorization and Identity management for EOSC users.
Creates PaaS deployments and provisions resources for
offer
EOSC
PaaS Orchestration and provisioning
[Software System]
Allows PaaS operators to manage PaaS deployments and resources.
register/update service & resources metadata
Cloud/HPC Providers
[Software System]
AI4EOSC
[Project]

[System Context] AI4EOSC Platform

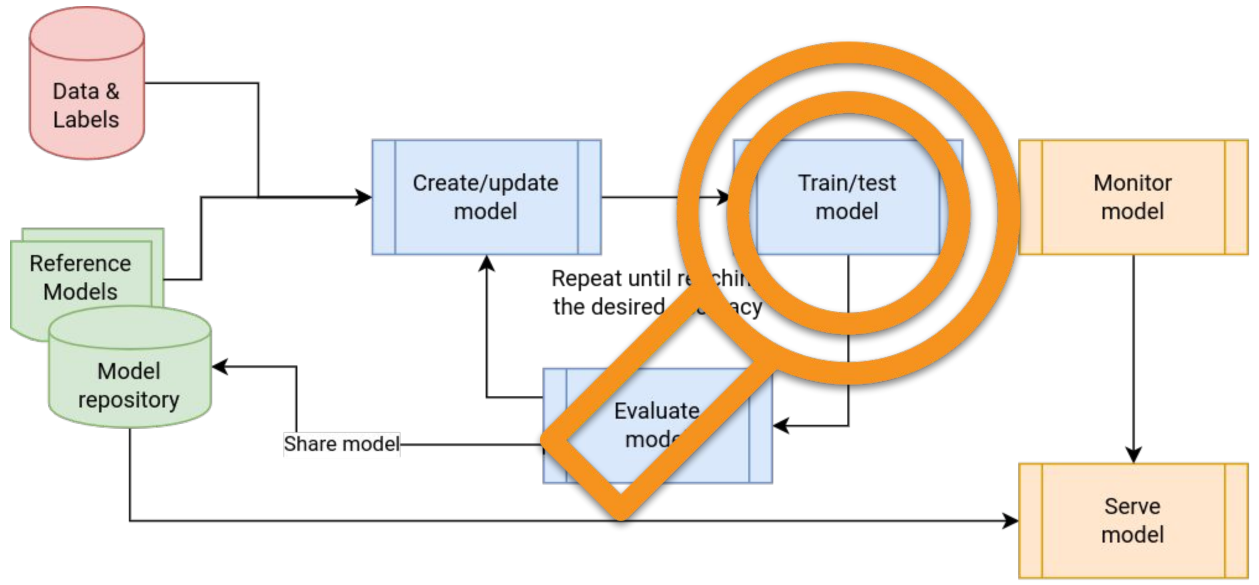
Tuesday, February 28, 2023 at 9:23 AM Coordinated Universal Time

iMagine

For users

Platform features

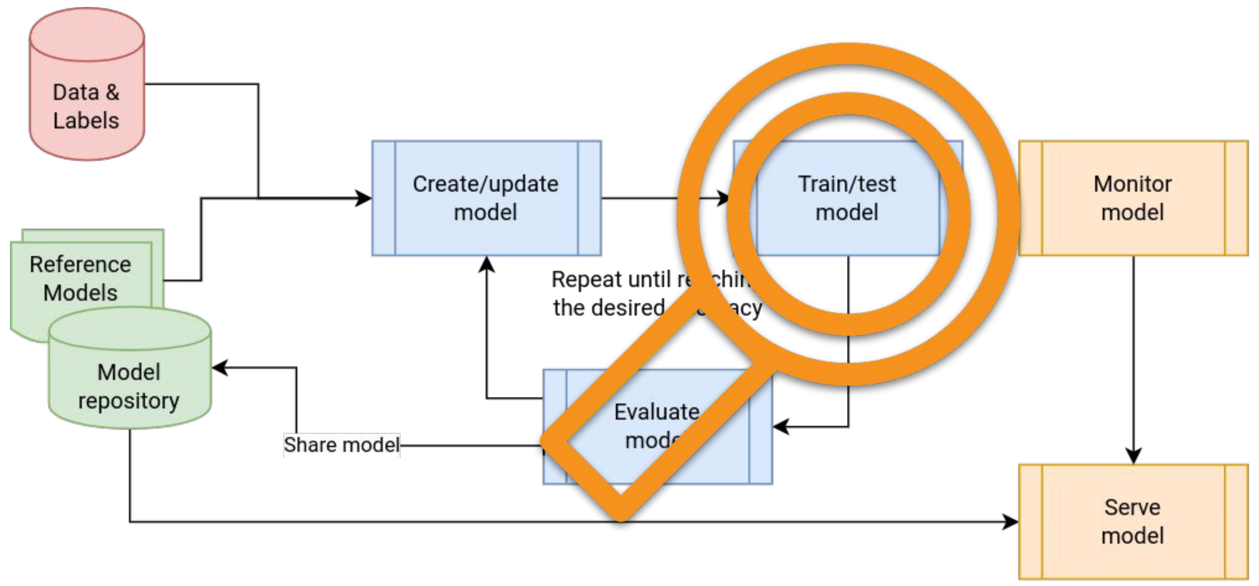
Training: upcoming features



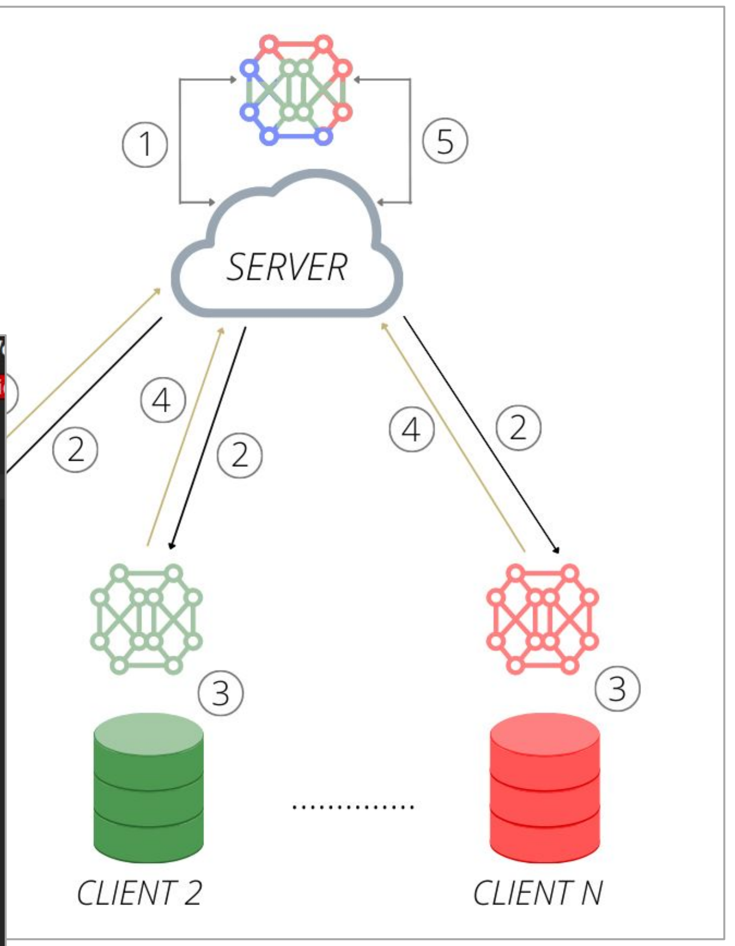
- Other distributed training schemes
    - Split learning
    - Parallel training with Horovod
    - TF Parameter server
- Model provenance system (MLFlow)
    - Including FAIR principles for ML models
- Experiment centric dashboard
    - i.e. group different models, data, trainings into one experiment
    - Allow to compare easily compare results
- Integration with additional online storage systems

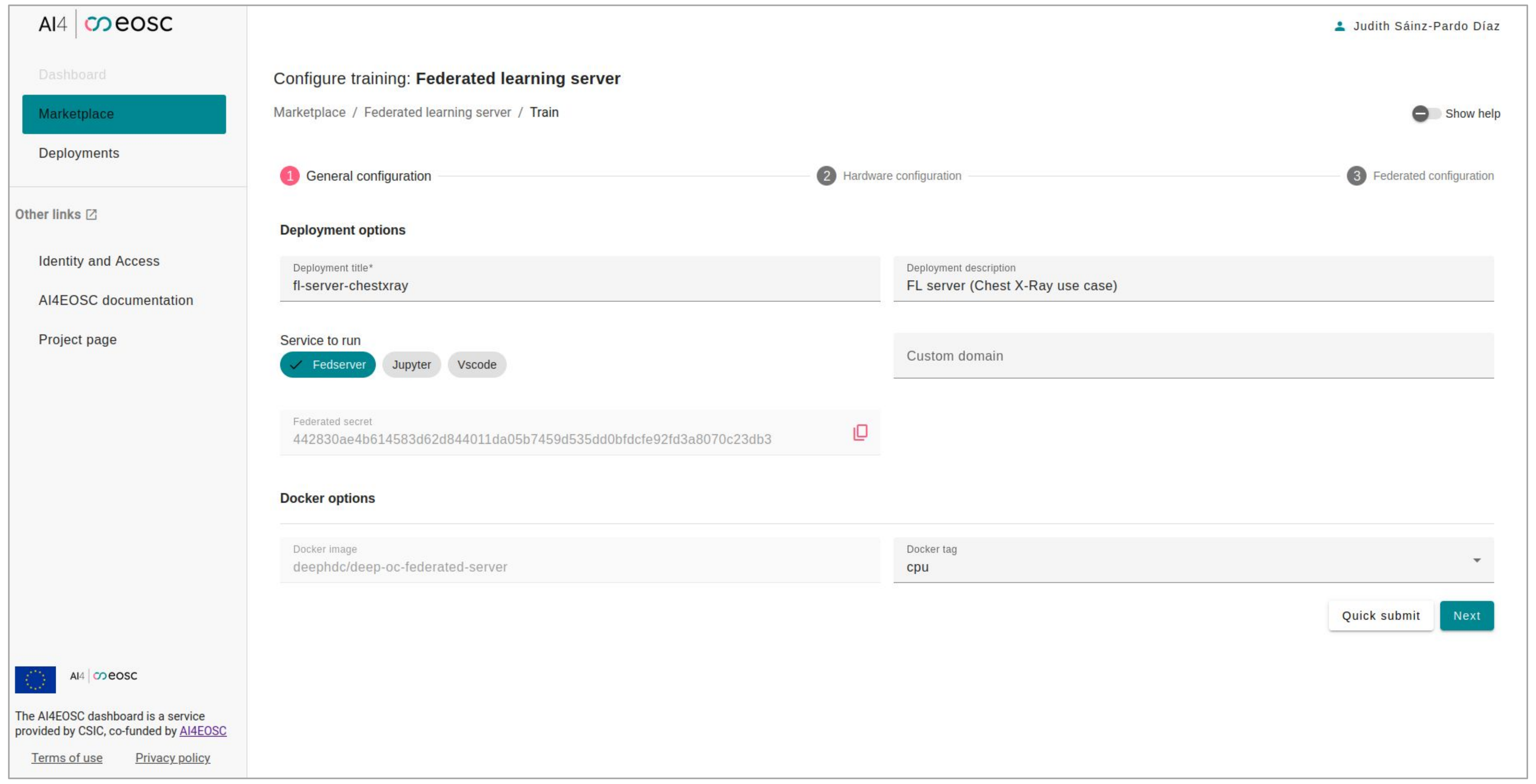
Services for AI/ML development

Training: federated learning



- Collaborative and decentralized approach to build ML models
    - No need to centralize a dataset (i.e. technical or privacy restrictions)
- Management of experiments through platform dashboard
- Participating clients both within AI4EOSC platform or external (with authentication)





Configure training: **Federated learning server**

Marketplace / Federated learning server / Train

1 General configuration — 2 Hardware configuration — 3 Federated configuration

Deployment options

Deployment title*: fl-server-chestxray

Deployment description: FL server (Chest X-Ray use case)

Service to run: ✓ Fedserver | Jupyter | Vscode

Custom domain

Federated secret: 442830ae4b614583d62d844011da05b7459d535dd0bfdcfe92fd3a8070c23db3

Docker options

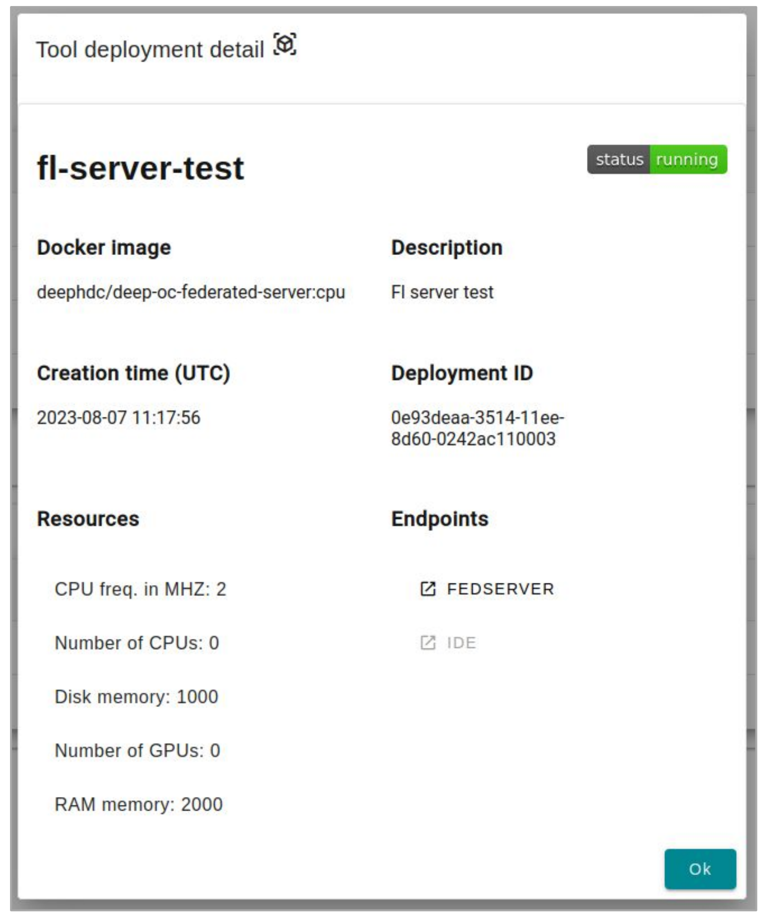
Docker image: deephdc/deep-oc-federated-server

Docker tag: cpu

Quick submit | Next

The AI4EOSC dashboard is a service provided by CSIC, co-funded by AI4EOSC

Terms of use   Privacy policy



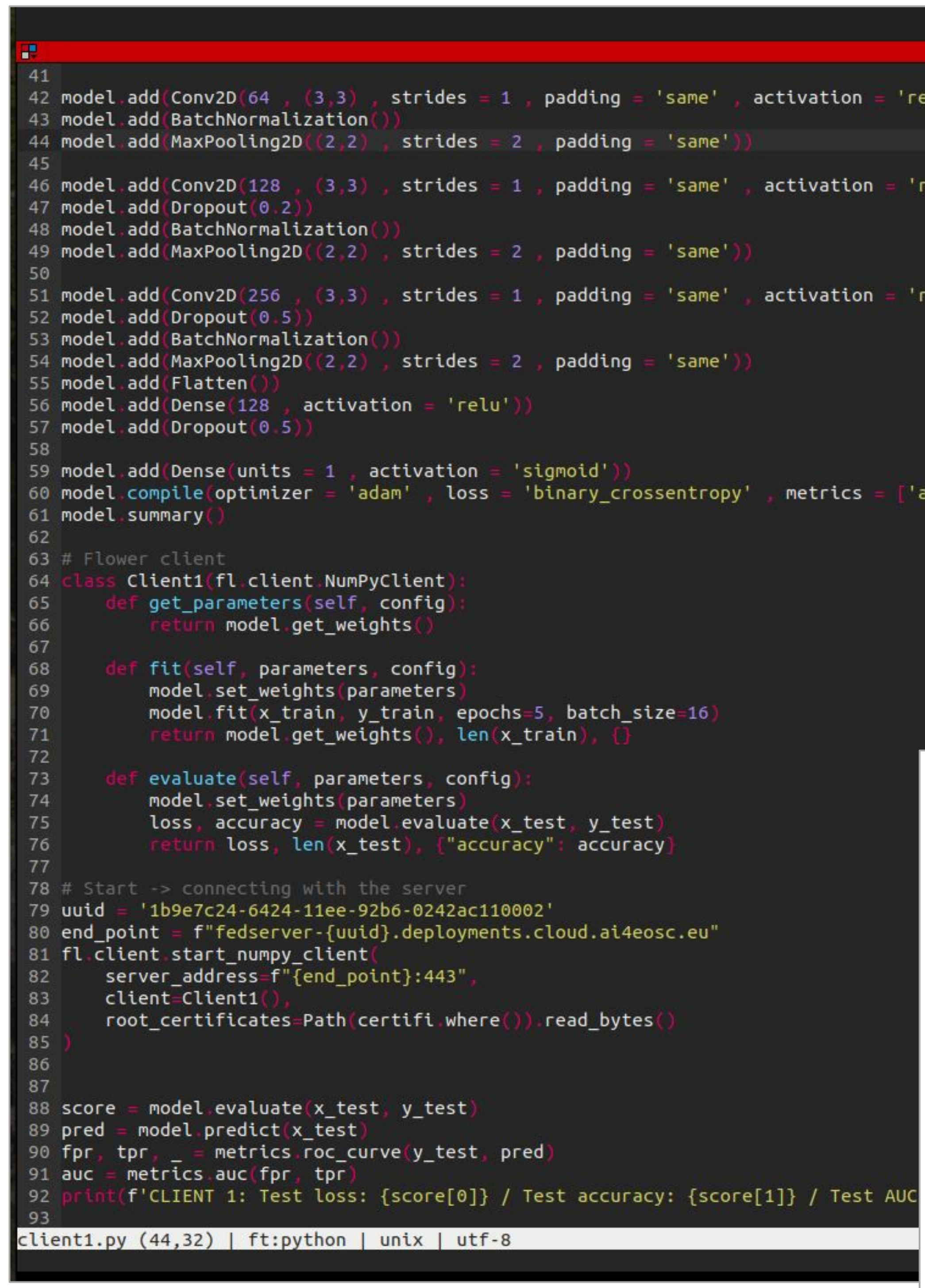
Tool deployment detail

fl-server-test

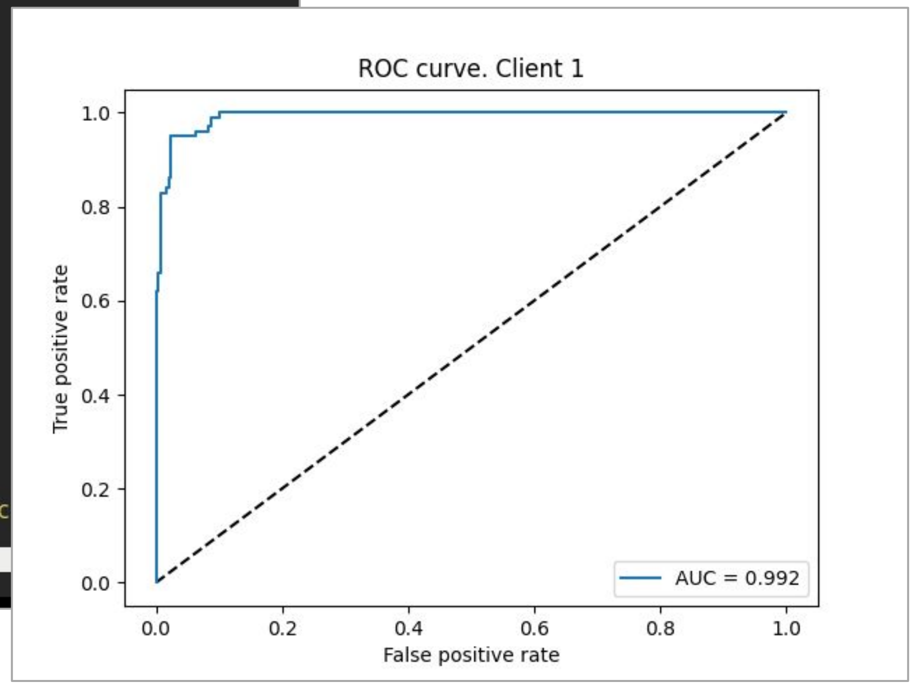
status running

Docker image	**Description**
deephdc/deep-oc-federated-server:cpu	Fl server test
Creation time (UTC)	**Deployment ID**
2023-08-07 11:17:56	0e93deaa-3514-11ee-8d60-0242ac110003
Resources	**Endpoints**
CPU freq. in MHZ: 2	FEDSERVER
Number of CPUs: 0	IDE
Disk memory: 1000	
Number of GPUs: 0	
RAM memory: 2000	

Ok

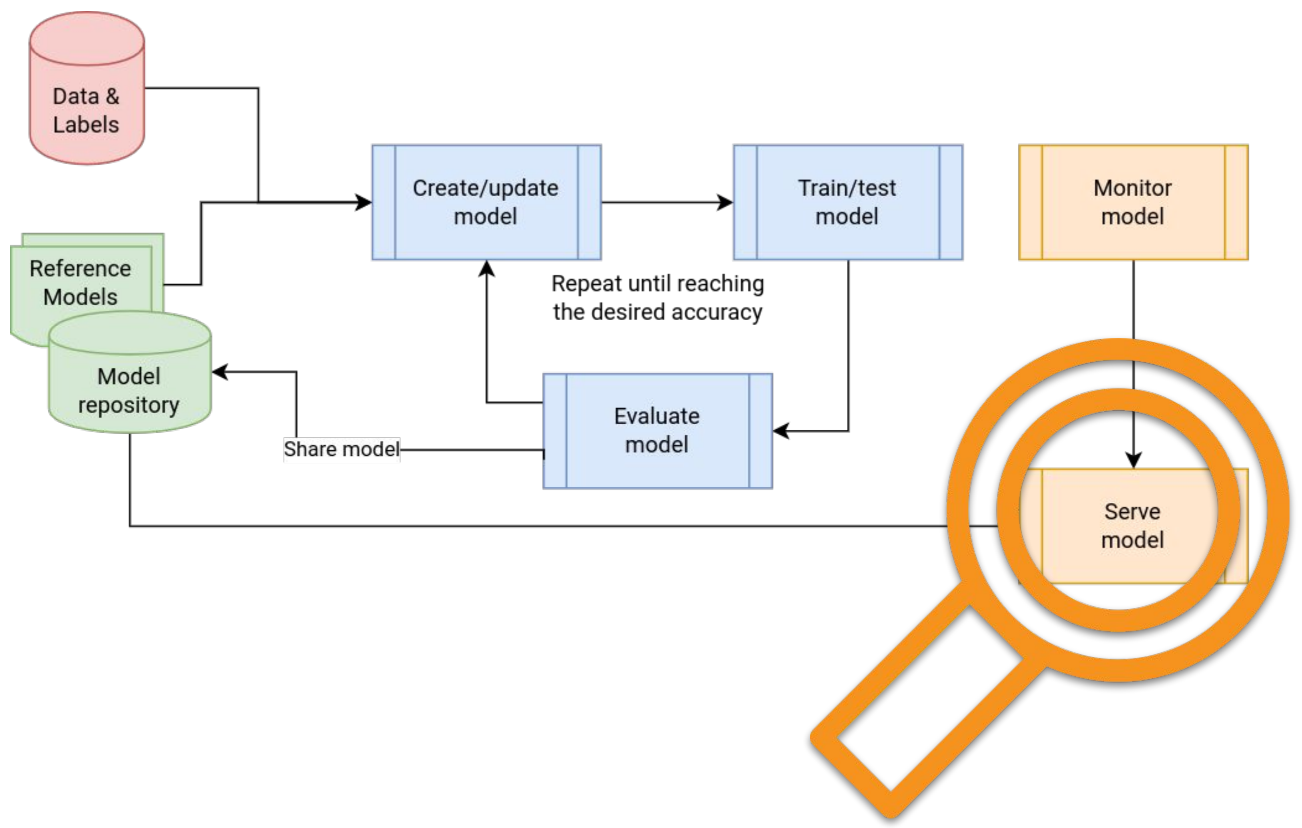


```
ubuntu@client1-fl: ~/c
ubuntu@client1-fl: ~/cli

model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2,2) , strides = 2 , padding = 'same'))

model.add(Conv2D(128 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPooling2D((2,2) , strides = 2 , padding = 'same'))

model.add(Conv2D(256 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.5))
model.add(BatchNormalization())
model.add(MaxPooling2D((2,2) , strides = 2 , padding = 'same'))
model.add(Flatten())
model.add(Dense(128 , activation = 'relu'))
model.add(Dropout(0.5))

model.add(Dense(units = 1 , activation = 'sigmoid'))
model.compile(optimizer = 'adam' , loss = 'binary_crossentropy' , metrics = ['accuracy'])
model.summary()

# Flower client
class Client1(fl.client.NumPyClient):
    def get_parameters(self, config):
        return model.get_weights()

    def fit(self, parameters, config):
        model.set_weights(parameters)
        model.fit(x_train, y_train, epochs=5, batch_size=16)
        return model.get_weights(), len(x_train), {}

    def evaluate(self, parameters, config):
        model.set_weights(parameters)
        loss, accuracy = model.evaluate(x_test, y_test)
        return loss, len(x_test), {"accuracy": accuracy}

# Start -> connecting with the server
uuid = '1b9e7c24-6424-11ee-92b6-0242ac110002'
end_point = f"fedserver-{uuid}.deployments.cloud.ai4eosc.eu"
fl.client.start_numpy_client(
    server_address=f"{end_point}:443",
    client=Client1(),
    root_certificates=Path(certifi.where()).read_bytes()
)


score = model.evaluate(x_test, y_test)
pred = model.predict(x_test)
fpr, tpr, _ = metrics.roc_curve(y_test, pred)
auc = metrics.auc(fpr, tpr)
print(f'CLIENT 1: Test loss: {score[0]} / Test accuracy: {score[1]} / Test AUC

client1.py (44,32) | ft:python | unix | utf-8
```

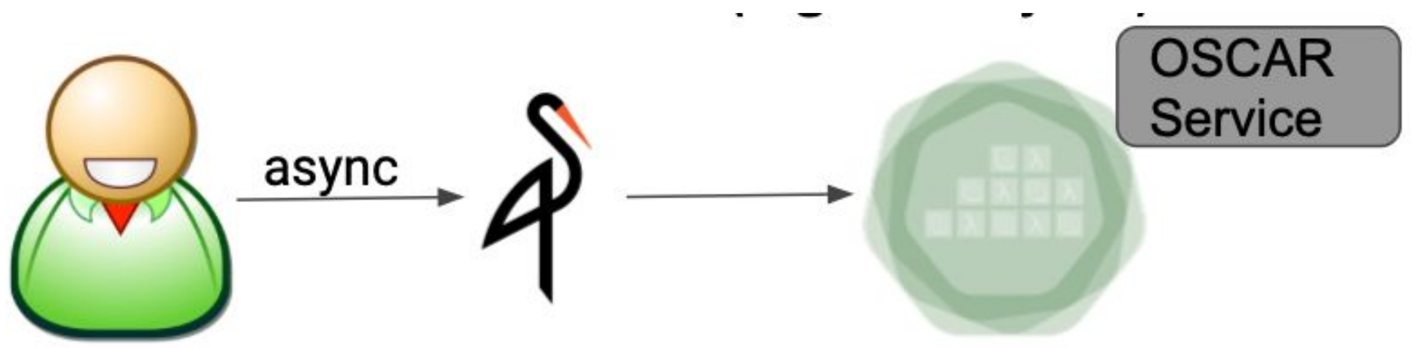


Services for AI/ML development

Deployment: OSCAR

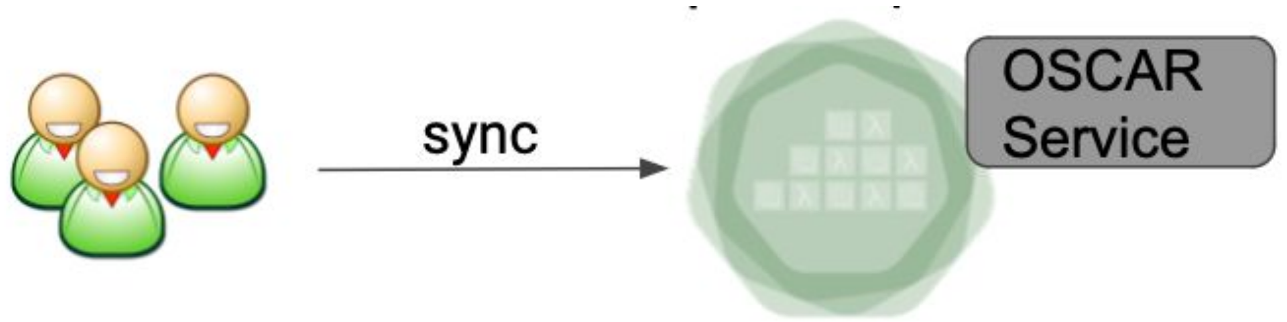


- OSCAR (https://oscar.grycap.net)
  to run the AI models for inference (AI as a Service)
  - Serverless event-driven execution
    - Asynchronous Mode: Files uploaded to the object-store trigger the invocation of a data-processing script that is run inside a container (out of user-defined Docker image) within a scalable Kubernetes cluster (e.g. batch jobs)



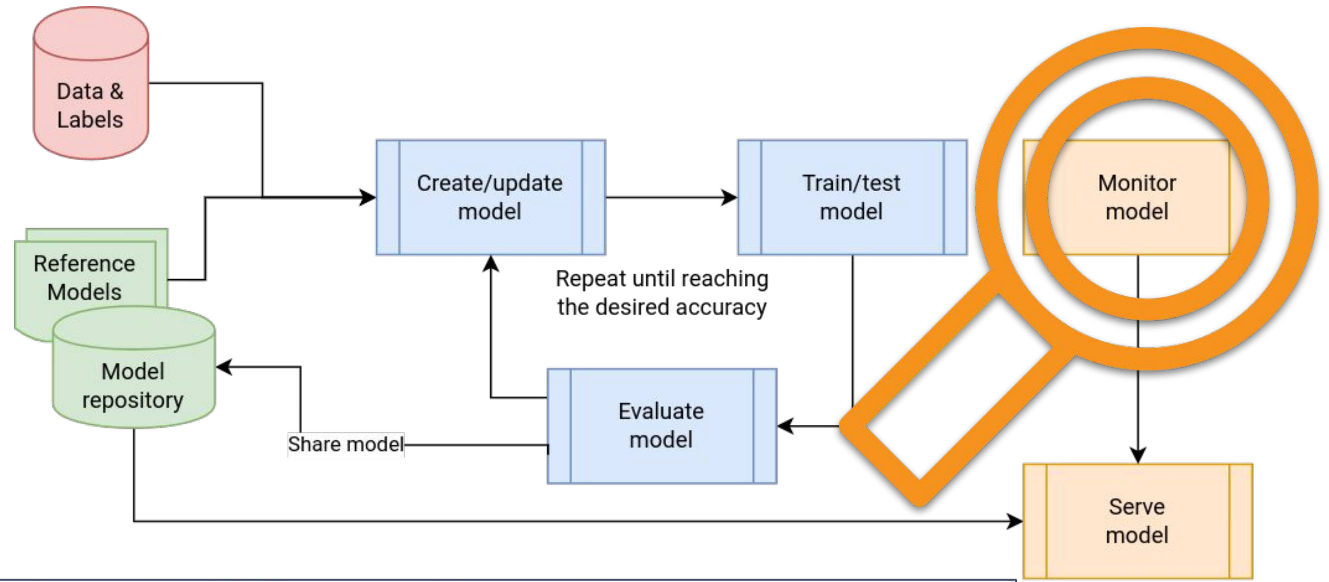
    - Synchronous mode: Scalable HTTP-based endpoints (based on KNative)

- https://inference.cloud.ai4eosc.eu/ui/
- https://inference.cloud.imagine-ai.eu/ui/

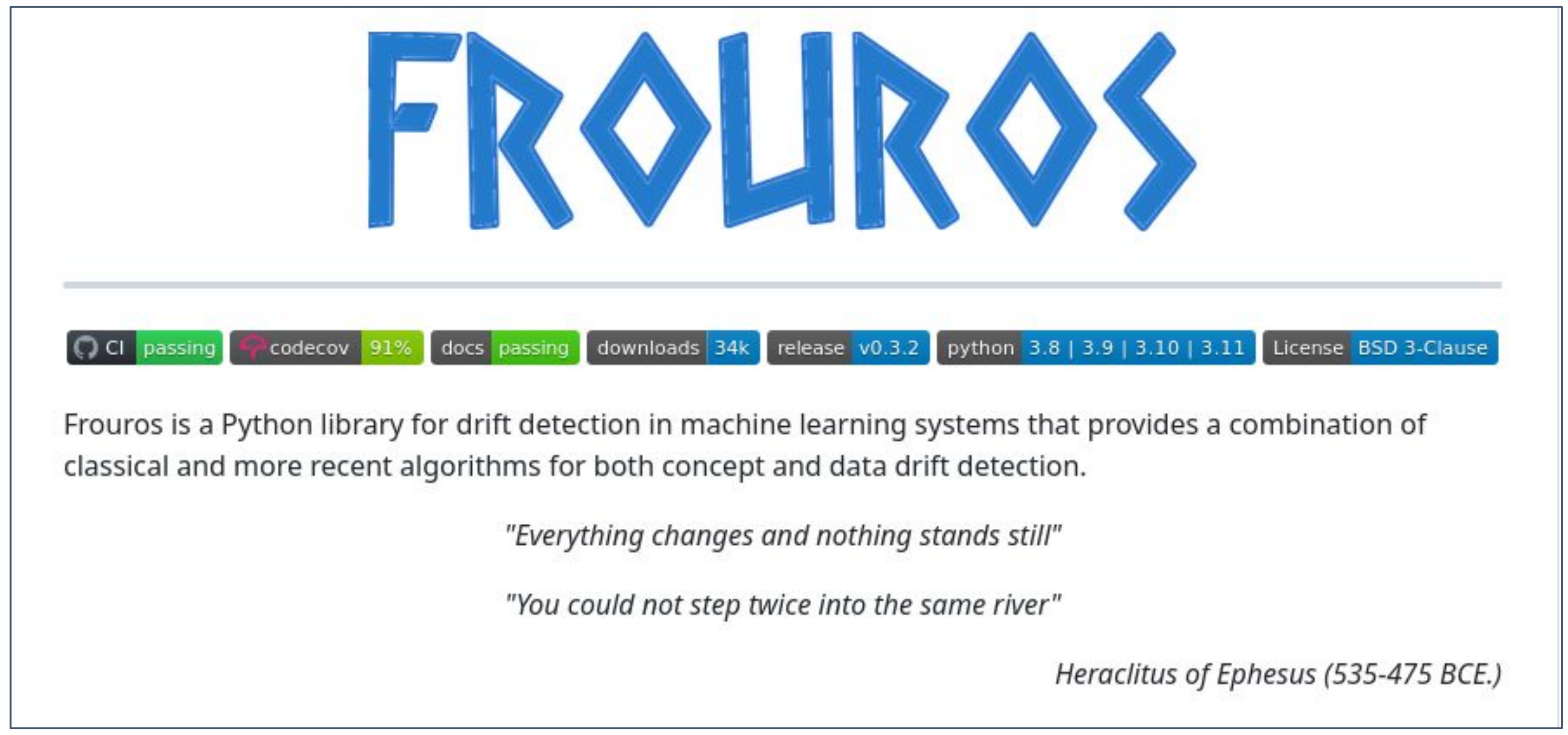


Services for AI/ML development

Deployment: Drift detection



- Monitoring of models in production is not enough
  - Model learns from data, data is not stationary
  - Concept learnt by them model may change over time
- Data and concept drift detection → essential to build more robust models
- Frouros: state-of-the-art library for drift detection in ML problems
  - https://github.com/IFCA/frouros
- Ongoing work towards online services for drift detection → MLOps pipelines with drift detection

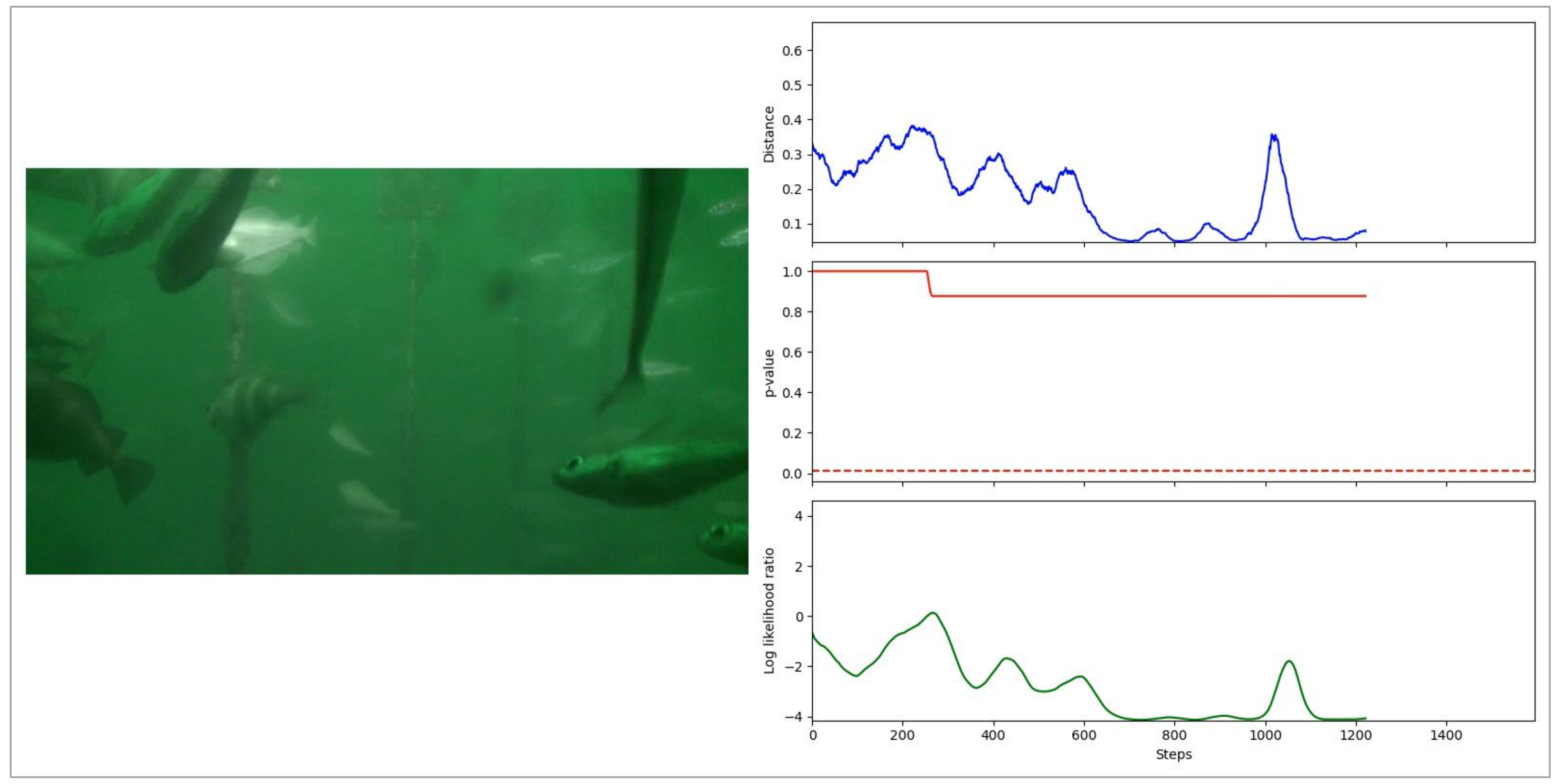


FROUROS

CI passing | codecov 91% | docs passing | downloads 34k | release v0.3.2 | python 3.8 | 3.9 | 3.10 | 3.11 | License BSD 3-Clause

Frouros is a Python library for drift detection in machine learning systems that provides a combination of classical and more recent algorithms for both concept and data drift detection.

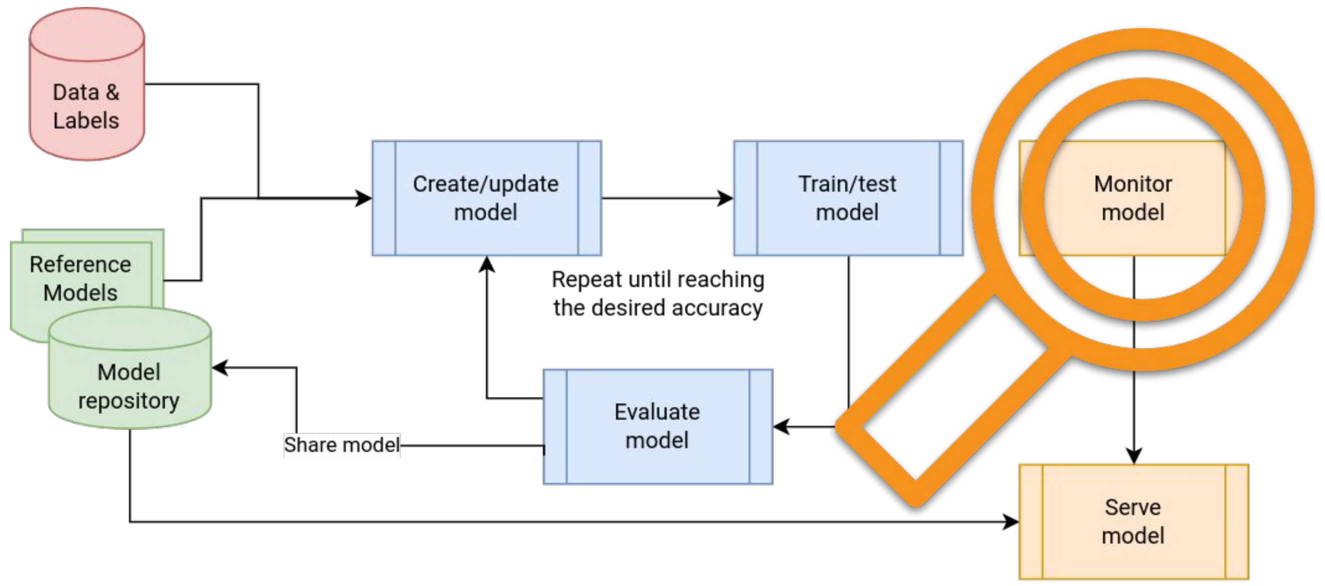
> *"Everything changes and nothing stands still"*
>
> *"You could not step twice into the same river"*
>
> *Heraclitus of Ephesus (535-475 BCE.)*



Example: data drift detection in underwater video

Services for AI/ML development

Deploy and monitor: upcoming



- Streamlining deployment and monitoring of models through CI/CD for ML applications and services
- MLOps pipelines definition
  - Automation for the whole ML lifecycle
  - Retraining based on given events (e.g. low accuracy)
- Compatibility with other ML deployment frameworks
- Improvements in API model definition (i.e. DEEPaaS)